ADMINISTRACION DE REDES GNU/LINUX

GUIA DE ESTUDIO HACIA UNA CAPACITACION SEGURA

FUNDACION

Código Libre Dominicano

Antonio Perpinan

Fundación Código Libre Dominicano
http://www.codigolibre.org
e-mail: info@codigolibre.org

(1) Padre Pina No. 102, Zona Universitaria,
Santo Domingo
República Dominicana

(2) Calle Estrella Sadhalá No. 5
Jardines Metropolitanos,
Santiago
República Dominicana

Diagramación y Diseño de portada:
Nicaury Benítez Cortorreal

ISBN
88-99999-99-9

Depósito legal:
b8888.99

Impreso y encuadernado por
IMPRESOS GAMMA
Calle #12
Los Ríos
Santo Domingo-República Dominicana

Impreso en República Dominicana
Printed in The Dominican Republic

ADMINISTRACION DE REDES GNU/LINUX

GUIA DE AUTO ESTUDIO HACIA UNA CAPACITACION SEGURA

INTRODUCCIÓN

LOS PROPÓSITOS DEL CURSO

Los profesionales de la tecnología de la información (TI) son críticos hoy día para el ambiente de negocio. Adquirir las herramientas y conocimiento disponible en la tecnología de hoy es vital. GNU/Linux y el Código Libre y Abierto han colocado un nuevo estándar en lo que es desarrollo e implementación de aplicaciones nuevas y personalizables. GNU/Linux continúa ganando espacio de reconocimiento entre los profesionales y administradores del TI debido a su flexibilidad, estabilidad, y su poderosa funcionalidad. A medida que más empresas utilizan GNU/Linux, crece la necesidad de soporte y planificación sobre la integración de GNU/Linux en infraestructuras nuevas y/o existentes. El rol del administrador es guiar la implementación y desarrollo de soluciones basadas en GNU/Linux. Su éxito o derrota dependerán de su conocimiento y experiencia de esta fantástica arquitectura.

Este curso es un repaso comprensivo de las características y funcionalidad de GNU/Linux, orientada a preparar al estudiante con las herramientas necesaria para la certificación. Explicación detallada se provee de los conceptos claves, muchos conceptos y utilidades de GNU/Linux son idénticos sin importar la distribución específica siendo utilizada. Algunas características están disponibles en algunas distribuciones, y otras son añadidas durante la instalación. La naturaleza de GNU/Linux y el Software Open Source, es tal, que cambios al fuente y cambio a funcionalidad de cualquier componente debe ser incluido en la distribución específica . Los conceptos sublime de las capacidades de GNU/Linux se mantienen consistentes a través de cada distribución, kernel y cambio de Software.

Este curso ha sido desarrollado de acuerdo con los estándares de la industria de certificación de GNU/Linux. Los objetivo de la certificación de Sair Linux y GNU han sido elementos claves en el desarrollo de este material. La secuencia de los exámenes de certificación Sair/Linux provee la gama más amplia de los conceptos necesario para dominar GNU/Linux. Los objetivos de las certificaciones LPI y RHCE también son incluidos. El CD interactivo y la página Web con el curso contiene videos digital y pequeñas prácticas de de selección múltiples igual a los del examen. En el libro LA GUIA DEL ESTUDIANTE se provee una guía específica para la preparación de la certificación.

Este libro provee los conceptos y principios fundamentales necesarios para administrar un sistema GNU/Linux. Los conceptos y las tareas de administración pueden ser un poco amplios. Se le dará una explicación del rol del administrador, estructura y función detallada del kernel, y cubriremos tópicos administrativos claves del manejo de paquetes, procesos, espacio de disco, Backups y los usuarios así como las tareas programáticas, y los Logs/Registros del sistema. Este conjunto de herramientas te permitirán apropiadamente administrar un sistema GNU/Linux sea este de unos cuantos hasta miles de usuarios. Estos capítulos también te proveerán la información que necesitas para Certificarte.

Fundamentos de GNU/Linux proporciona una introducción a profundidad de los conceptos y de los principios que son necesarios para instalar un sistema GNU/Linux y desenvolverse en los ambientes de ventana del X y de la línea de comandos. Este manual da la dirección paso a paso para las distribuciones importantes de GNU/Linux y su instalación, incluyendo RedHat, Debian, Mandrake y Slackware. Se enfatizan los conceptos de instalación, las utilidades, y la funcionalidad de GNU/Linux común a todas las distribuciones y estas se explican en detalle adicional. Un principiante o un experto pueden aprender o repasar los conceptos de particionar discos y localizar los archivos de configuración, usando el shell y las consolas, crear los scripts, y editar archivos de texto que permanecen dominantes, sin importar la nuevas herramientas gráfica, para los ajustes de configuración. Este conjunto de tópicos permitirá que usted instale y configure correctamente un sistema GNU/Linux. En estos capítulos también se le provee la información necesaria para certificar sus habilidades en GNU/Linux

METAS DEL CURSO

Este curso le proveerá con la información que necesitas para completar lo siguientes tópicos:

- Describir los componentes estructurales y distinguir entre una distribución de GNU/Linux y otra.
- Describir software de fuente abierta (Software Open Source) y entre GNU/GPL.
- Crear los disquetes de arranque de instalación.
- Instalar las principales distribuciones de GNU/Linux: RedHat (RPM), Debian (DPKG) y Slackware (tar.gz).
- Utilizar los ambientes de escritorio KDE y GNOME.
- Instalar y configurar XFree86.
- Localizar y utilizar la ayuda en línea.
- Configurar el hardware del sistema.
- El uso de fdisk o el cfdisk para crear, corregir, y suprimir particiones del disco.
- Utilizar el LILO/GRUB para manejar opciones para cargar el sistema.
- Arrancar el sistema, cambiar los runlevels, y cerrar o re-iniciar el sistema.
- Utilizar los disquetes de rescate para iniciar un sistema que se ha dañado.
- Describir el sistema de archivos jerárquico de GNU/Linux y el papel de los directorios y archivos claves en la organización del sistema. Trabajar con eficacia en la línea de comando de Linux usando comandos comunes del shell, streams, tuberías, filtros, y cambio de dirección.
- Usar scripts del shell para realizar tareas repetitivas rápidamente.
- Abrir, corregir, y almacenar documentos de texto usando el editor ‘vi’.
- Manejar los sistemas de impresión locales.
- Describir algunas aplicaciones comunes disponibles al usuario para sus tareas, tales como: navegar en Internet y acceso a E-mail, procesamiento de textos, presentaciones, hojas de cálculo, y manejo de gráficos.

EJERCICIOS

Los ejercicios en este manual son diseñados para dar practicas reales en los ambientes de redes y aislados (stand-alone o networking) al usuario. Es altamente recomendado que usted complete todos los ejercicios en cada capítulo antes de continuar al próximo. Entendemos que en raros casos tal vez esto no sea conveniente cuando estudia fuera del taller. Si por alguna razón no puedes completar un ejercicio por circunstancias ajenas, debes planificar completarlo tan pronto sea posible.

Existirán ejercicios que no podrás completar por el limitante de equipo ó software. No permita que esto le impida completar los otros ejercicios que resten en el capítulo ó módulo.

TOME NOTA

Los ejercicios en este libro fueron diseñados para ser ejecutados en un equipo de prueba y nunca deben ser llevados acabo en uno trabajando y donde se ejecuten aplicaciones importantes. Instalar GNU/Linux, reparticionar para instalar GNU/Linux, o practicando los ejercicios en una LAN u ordenador de trabajo puede causar problemas de configuración, lo cual puede conllevar a perdidas irreparable de data y dispositivos periféricos. Por favor siempre recuerde esta advertencia. Es preferible que dediques una estación de trabajo para practicar estos ejercicios. Instalar GNU/Linux en una situación dual-boot es una alternativa razonable, pero aún así conlleva ciertos riesgos.

WEB y CD

Una parte muy clave de esta serie de auto-aprendizaje es el portal de soporte. Las lecciones que le indiquen visitar la página web o el CD-ROM que le acompaña, a menudo, es para ayuda con los conceptos que son mejor entendidos después de una descripción visual. Los segmentos video Digital proporcionan una ilustración gráfica acompañada por una narración de los instructores. Estas lecciones son ideales, ambos como introducciones para afinar conceptos y para ayudar el refuerzo.

RECUERDE

Como herramienta de soporte les ofrecemos el CD interactivo, incluido en este libro, y nuestra página web http://www.abiertos.org y allí accesar hacia la sección Linux-Certificación, estos contienen exámenes de prueba de las diferentes certificaciones. Recreamos el escenario de las preguntas de selección múltiples, multi-selección y falso verdadero. Es muy importante que tome muchas horas de practicas antes de intentar pasar el examen de certificación que le corresponda ya sea LPI ó RHCE.

CAPITULO 1

REDES DEL SISTEMA OPERATIVO GNU/LINUX

FUNDAMENTOS PROTOCOLO DE INTERNET (IP)

OBJETIVOS

Al completar este capítulo usted podrá:

- Describa el proceso y los elementos claves de la configuración básica de una red.
- Describa la tecnología básica que son el motor de las redes de Internet, Ethernet y Redes de Area.
- Liste y describa la implementación de las interfases más comunes de redes.

PREGUNTAS PRE-EXAMEN

Las repuestas se encuentran en el Apéndice A.

1. Defina una red.
2. ¿Por qué usar una red?
3. Liste los niveles del modelo de cuatro capa del TCP/IP, luego liste 2 protocolos usados por cada uno.

INTRODUCCIÓN

Una Red (Networking) se define como dos ó más computadores conectadas entre si para compartir data. El trabajo en red permite que la información sea distribuida fácil y rápidamente a través de un sistema de protocolo, cables, y hardware. Usted puede hoy día utilizar tecnología alternativas como lo son, infra-roja ó laser, para permitirle la comunicación entre dos ó más computadoras.

Las redes son extremadamente popular hoy en día ya que ellas permiten que los usuarios compartan su data en una manera eficiente. Ya es cosa del pasado que los usuarios tengan que colocar su información en disquetes o imprimirlo para poder físicamente transportar su data de un lugar a otro. No existe mejor método alguno que la redes para organizar y facilitar el trabajo de el trabajo en grupo..

Como su nombre lo indica que una red son dos ó más computadoras conectadas entre si vía un medio físico y protocolos de comunicación, una red puede ser un negocio pequeño que se desenvuelve en una pequeña area, la cual se denominan Redes de Area Local (Local Area Network) ó LAN. Una red también puede conectar varias LAN a través de una larga distancia ó una Red de Area Extendida. También son conocidas como WAN (Wide Area Network). Es posible extender una serie de WAN colocadas en diferentes partes del mundo interconectados como una sola red permitiendo que millones de usuarios compartan sus datos, así como es el Internet. Antes de poder comprender el Internet, internetworking, y el TCP/IP, usted está en la obligación de entender como funcionan las redes tradicionales.

El TCP/IP permite que computadores de diferentes marcas, modelos, sistemas operativos y capacidades (desde un mainframe hasta una PC de escritorio) se comuniquen. Desde su adopción en el 1983 por las grandes redes que conformaron el Internet, el TCP/IP ha superado con crece todas las expectativas que se tenía de el. Hoy día es el conjunto de protocolo de red más usado en el mundo y es el protocolo que hace posible el Internet, el WAN del mundo.

En este capítulo discutiremos la historia del Internet, los modelos que usa el Internet para comunicarse, la arquitectura del Internet, los protocolos comunes usados por en el Internet, y los protocolos seriales de enlace (serial link protocols). También discutiremos y analizaremos los documentos de Requisición de Comentarios ó RFC (Request for Comments RFC), documentos que definen y sirven de referencia para los protocolos de Internet.

CONCEPTOS DE REDES/NETWORKING

Existen varios conceptos fundamentales que son básicos al analizar ó diseñar una red. El concepto central y fundamental de una red es que permita la comunicación y la compartición de recursos entre dos o más computadoras. Los detalles de como esta dinámica ocurre y los elementos que diferencia una red serán discutidos en esta sección:

- Redes Tradicionales
- Redes de Internet/Internetworking
- Ambiente de Redes
- Modelos de Red
- Categorías de Red
- Topologías de Redes
- Sistemas Operativos de Red
- Servicios Básicos de Red

Redes Tradicionales

Tradicionalmente cuando una compañía elige una red, esta trata de que la red que elige y luego use sea de un solo fabricante. Este tipo de elección creaba las redes homogéneas y centradas a una marca de fabricante en específico. La gran mayoría de organizaciones elegían vendedores como IBM, Novell, ó Banyan, para que le proveyeran soluciones de redes ya que redes compuestas por equipos de un sólo suplidor aseguran un mínimo de entrenamiento y capacitación para los empleados y sus profesionales de TI. El razonamiento era que utilizando el mismo tipo de redes creaba una situación lo más simple posible.

En el desarrollo de sus carrera puede ser que usted confronte mucho tipos de redes diferentes como las que hacemos aquí cita. Tiempo atrás el caso hubiese sido totalmente diferente y en sus labores lo más probable es que hubiese encontrado sólo un tipo de redes de compañía a compañía. Así es que hoy día debe tratar de familiarizarse con diferentes tipos de redes y también los protocolos en la diferentes situaciones que se le presenten, después de aprender este nuevo protocolo y topología, lo más probable que en el pasado hubiese sido todo lo que tenia que dominar para llevar acabo las tareas que necesita ejecutar.

Redes de Internet/Internetworking

Desde los años 90, han ocurrido cambios fundamentales sobre lo que redes concierne. Hoy en día muchos tipos diferentes de redes pueden trabajar interconectadas. Esta necesidad nace por causa de que organizaciones y sus divisiones necesitan estar permanentemente conectadas para compartir su información y recursos. La tarea de trabajar sistema diferentes en una manera heterogénea, como se trabaja interconectado a través del Internet, es conocido como internetworking.

Este tipo de trabajo en red representa un cambio dramático, ya que en el esquema tradicional de soluciones de redes, una organización sólo podía comunicarse consigo misma, pero para hacerlo con otras organizaciones, esta tenía que utilizar los métodos tradicionales, como lo fué tradicionalmente el correo. La motivación detrás de esta interconexión de las organizaciones y sus dependencia es causada por la necesidad de la transferencia continúa de información a través de extensas areas geográficas a muy alta velocidad. Hoy día será muy común que usted tenga que interconectar diferentes tipos de redes bajo una sola unidad lógica la cuáles pueden serán accesadas a través de la red pública, el Internet.

Ambientes de Red

Hoy en día existen dos principales ambiente de red en uso hoy: Redes de Area Local (LANs) y Redes de Area Amplia (WANs).

Redes de Area Local (LANs)

La definición más general de una red de área local (Local Area Network, LAN), es la de una red de comunicaciones utilizada por una sola organización a través de una distancia limitada, la cual permite a los usuarios compartir información y recursos como: espacio en disco duro, impresoras, CD-ROM, etc.

Redes de Area Amplia (WANs)

Es una red comúnmente compuesta por varias LANs interconectadas y se encuentran en una amplia área geográfica. Estas LAN's que componen la WAN se encuentran interconectadas por medio de líneas de teléfono, fibra óptica o por enlaces aéreos como satélites.

Entre las WAN's más grandes se encuentran: la ARPANET, que fué creada por la Secretaría de Defensa de los Estados Unidos y se convirtió en lo que es actualmente la WAN mundial: INTERNET, a la cual se conectan actualmente miles de redes universitarias, de gobierno, corporativas y de investigación.

Tipos de Redes

En sus orígenes las redes operaban con modelo centralizado, donde un mainframe suplía todos los recursos a computadoras denominadas estaciones de trabajo. Comúnmente esto limitaba sólo a redes de area amplia, que podían ser costeadas por grandes Universidades y compañías alto capital de inversión. A mediado de los años 80, la mayoría de estos negocios adoptaron el modelo cliente/servidor. En ves de usar servidores centrales del tipo mainframe, el modelo cliente/servidor utiliza un modelo modular que permite a pequeñas y medianas empresas utilizar estas soluciones de redes. Con la llegada del Internet, se pudo superar otra etapa de los modelos de red, que fué cuando estas pequeñas y medianas compañías pudieron utilizar económicamente modelos de redes de area amplia a través del uso de aplicaciones basado en la Web.

Mainframes

Sistemas Mainframe, también conocido computación centraliza, proveyeron la primera solución de rede practicas. Este método centralizado de emplear servidores centrales, o mainframe y terminales remotas. Normalmente estas terminales denominadas tontas o sin discos(diskless) solo pueden requerir información y no procesarla. El procesamiento de la información ocurre del lado del servidor.

Tradicionalmente obtener información desde el mainframe involucra un alto procesamiento por esta. Cuando un terminal pide información, esta requisición es enviada al mainframe. el mainframe procesa la requisición y obtiene la información deseada desde la base de datos o cualquier otra fuente. después que el procesamiento a culminado, el mainframe estructura y reforma la información al terminal. Veremos más adelante como el modelo cliente servidor difiere a este modelo.

El modelo de computación tipo mainframe tiene dos desventajas principales. La primera es que el mainframe efectúa todas las tareas de procesamiento. Las segunda es que los paquetes de petición y respuesta enviados entre el terminal y el mainframe ocupan mucho ancho de banda de la red. Estas dos desventajas en redes amplias y de mucho uso causan congestiones de red no aceptables.

Debido a la gran inversión que significan los mainframe en instituciones como, la banca, telefónicas, universidades, etc este modelo de mainframe es prevalente y lo será por mucho tiempo. Con el crecimiento de las tecnologías basada en la Web, las interfases basada en la web y otras tecnologías emergentes han reemplazado o por lo menos desplazado muchas de las plazas donde tradicionalmente se usaba un ambiente de mainframe y terminales.

Modelo Cliente/Servidor

La gran mayoría de aplicaciones en uso hoy día están basadas en el modelo cliente/servidor. En este modelo, la funciones se dividen entre los dos componentes de una aplicación. Basado en la concepción de que una parte proporciona el servicio (el servidor) y la otra usando el servicio (el cliente). Ejemplos de un servicio incluye acceso a una impresora, acceso a archivos almacenados en un sistema, ó acceso a una base de datos. Todos los servicios TCP/IP están basado en el modelo cliente/servidor. Normalmente, un servidor puede administrar las peticiones de multiple clientes.

El modelo cliente/servidor también es conocido como sistemas distribuidos, el cual reduce las congestiones de la red a través de una división de las tareas de procesamiento entre el cliente y el servidor. El servidor generalmente es más potente que que la computadora ejecutando el cliente y es responsable de almacenar y presentar la información.

En la mayoría de los ambientes de redes, existen varios Servicios disponibles. Es muy común que un computador sea un servidor para una aplicación y un cliente para otra. En una situación un computador puede ser un servidor de archivos, pero al momento de necesitar imprimir requiere de un servidor de impresión para que le lleve acabo esta función.

Otro buen ejemplo es sobre una WAN. Todos los que han navegado el World Wide Web (WWW) ha experimentado con aplicaciones cliente/servidor a través del suite de protocolo TCP/IP. Para accesar el WWW, el usuario ejecuta la aplicación cliente (en este caso el navegador Web, E.j.,., Mozilla, Firefox, Konqueror). Cuando el usuario desea accesar una página Web, este cliente contacta el servidor de páginas Web correcto sobre un enlace de red de área ampliada y nos despliega la página que el servidor Web nos envía.

En una sesión típica de navegación cuando un usuario sigue de un enlace a otro en un sitio web distinto, un cliente único contacto varios servidores. Al mismo tiempo, la máquina del usuario puede estar sirviendo también páginas Web con un servidor que están siendo accesadas al mismo tiempo por otros usuarios.

Una sesión típica Cliente/Servidor involucra el siguiente proceso:
1.- El usuario efectúa una petición de data.
2.- La computadora cliente traduce la petición en un formato que el servidor puede entender.
3.- El cliente envía la petición al servidor.
4.- El servidor procesa la petición, la cual puede involucrar comunicación con servidores remotos.
5- El servidor envía la repuesta al cliente.
6- El cliente envía la repuesta a su pantalla.

La diferencia clave entre este modelo de petición y repuesta y el que se presentó anteriormente usado por el mainframes es que el cliente procesa la mayoría de sus peticiones.

Además de compartir la responsabilidad de procesar las tareas, los beneficios de los clientes/servidores incluye una metodología modular del manejo del sistema. Debido a que el modelo cliente/servidor nos permite agregar nuevo componentes al sistemas, usted no esta limitado a una solo solución. Un tiempo atrás los ingenieros de sistemas tenían que elegir entre un sistema y el otro. Con la llegada de los Estándares abiertos como lo son el TCP/IP (Transmission Control Protocol/Internet Protocol), los sistemas heterogéneo pueden trabajar juntos más eficientemente. Por ejemplo, Servidores de UNiX y Windows NT que usan TCP/IP pueden trabajar juntos, permitiendole a los negocios escalar a soluciones de acuerdo con las demandas y necesidades del cliente. El modelo cliente/servidor es escalable porque este le proporciona la capacidad de ajustarse a nuevas demandas. El modelo Cliente/Servidor también concede a los usuarios más control sobre sus propios archivos.

Las organizaciones se encuentran cada día más y más en la necesidad de migrar hacia la arquitectura y aplicaciones basadas en el modelo cliente/servidor. Los sistemas de servidores y estaciones de trabajos basados en arquitecturas i386 (PC) brindan las mismas características encontradas en sistemas operativos de mainframe pero a un costo mucho más razonable para las pequeñas y medianas empresas. Se puede ir más lejo aún, existen hoy día miles de aplicaciones disponibles para estos sistemas operativos que se ejecutan sobre estas estaciones de trabajo y sus servidores.

La arquitectura Cliente/Servidor incorpora una flexibilidad considerable al distribuir los recursos en la red. Producto de la flexibilidad es la complejidad; la necesidad de entender los tipos de nodos de los clientes que son apropiados para las necesidades de su organización. Aún una selección más compleja es al escoger los nodos de servidores.

Las repuestas a muchas configuraciones del modelo cliente/servidor basados en:
- Los tipos de aplicaciones que caracterizan su ambiente de sistema
- Los usuarios y sus peticiones
- La arquitectura computacional y de redes

Redes Basadas en la Web

Redes basada en la arquitectura Web, también conocida como sistemas colaborativos, utiliza tanto tecnologías mainframe y cliente/servidor. Con el uso de TCP/IP, tecnologías de trabajo basado en intenet han permitido a las redes evolucionar a sistemas más y más decentralizados y distribuidos. En muchos de los casos, sistemas que se basan en la tecnologías Web representan la forma más radical del modelo computacional de tres capas. Redes altamente distribuidas, como son las extranets, las VPNs (Virtual Private Networks, Redes Privadas Virtuales), y Así sucesivamente, estas también dependen de navegadores de Internet para que sirvan de interfases de usuarios con mecanismo de servidores complejos.

El beneficio de este modelo es que puede combinar el poderío que representan los mainframe con la escabilidad del modelo cliente/servidor. Más aún, ya que las redes basadas en la tecnologías web dependen del uso de un navegador (browser), podemos presentar información sin la necesidad de programas especializados. Por esto es que redes de tecnología basada en la Web, proveen soluciones globalizadas que permiten a los usuarios obtener información y llevar acabo transacciones, independiente del sistema operativo que se encuentre y sin la necesidad de instalar programas adicionales. Las soluciones de comercio electrónico ó e-commerce, comprar carros, libros, etc, son un buen ejemplo de como se pueden llevar acabo transacciones seguras basadas en soluciones Web.

Categorizando las Redes

Toda red consiste en tres mismos elementos básicos:

Protocolos

Las reglas de comunicación que todos los elementos de una red deben estar de acuerdo. En el próximo capítulo discutiremos los protocolos de redes a fondo.

Medios Transmisión

El método por el cual todos los elementos de la red se interconectan. Se discutirá más adelante.

Servicios de Red

Recursos como son las impresoras, que deben ser compartidas con todos los usuarios de la red. Se discutirá más adelante.

Al margen de estas similitudes, existen dos tipos básicos de redes: peer to peer (persona a persona) y basadas en servidores. Existe una tercera arquitectura, redes empresariales (enterprise), la cual combina las dos.

Redes Peer-to-Peer

Una arquitectura de red peer-to-peer es aquella que no necesita de recursos dedicados, como es un servidor de archivos, además cualquier equipo (host) puede compartir sus recursos con otros sistemas en la red. En este tipo de redes, un mismo equipo puede funcionar como cliente y como servidor, ya sea haciendo peticiones de data ó sirviendola.

Por lo general, redes del tipo peer-to-peer tiende a resultar de menos inversión financieras y de menor requerimiento de conocimientos para quienes las operan que las redes del modelo cliente/servidor. Aunque estas redes son menos seguras y soportan menos usuarios (al rededor de diez ó menos es prudente), y experimentan más problemas con la administración de los sistemas de archivos debido a la falta de seguridad y control.

Ejemplos de productos de redes de PC del tipo peer-to-peer incluyen a:
- LANtastic
- Novell NetWare Lite
- Microsoft Windows for Workgroups (Para Grupos de Trabajos)
- Microsoft Windows 95/98

Redes Cliente/Servidor (Server-Based Network)

Una red basada en servidores es una configuración de nodos de los cuáles algunos están dedicados a proveer recursos a los otros nodos en la red denominados hosts. Los nodos dedicados que ofertan sus recursos son denominados Servidores. Estos recursos incluyen impresoras, aplicaciones, y documentos.

Redes Basadas en Servidores ofrecen seguridad a los usuarios ya que se puede mantener un inventario en una base de datos central de los usuarios y que recursos ellos pueden tener acceso. En la otra mana, los servidores dedicados suelen a ser un poco más costosos y requieren un administrador de redes a menudo a tiempo completo.

Algunos ejemplos de nodos de servidores comunes son:
- Servidores de Impresión
- Servidores de Archivos
- Servidores de Correo
- Servidores de Páginas Web

Nodos Cliente pueden accesar estos recursos a través de la red. Ejemplos de redes del modelo cliente/servidor incluyen:
- Novell NetWare
- UNiX
- Microsoft Windows NT

Redes Empresariales (Enterprise Network)

Redes empresariales proveen conectividad entre todos los nodos de una organización, sin importar localización geográfica, y son capaces de ejecutar las aplicaciones de misión criticas. Estas redes incluyen ambos modelos que hemos discutido hasta el momento peer-to-peer y cliente-servidor.

Una red empresarial puede consistir de multiple pila de protocolos (stack) y arquitecturas de redes. Aquí le presentamos algunas de las características de las redes empresariales:
- Los sistemas de la red son capaces de trasladar paquetes de información desde una arquitectura a la otra (llamados gateways ó pasarelas).
- Los sistemas que soportan múltiples arquitecturas están presentes en la red (sistemas multi-protocolar).

Las Pasarelas ó Gateways, arquitecturas de redes y sistemas de multiprotocolos serán discutidos más adelante.

Topologías de Redes

Topologías son configuraciones básicas que el personal que manejan sistemas de información utilizan para cablear las redes. Ellas son el diseño básico de cualquier red. Las diferentes topologías usadas para conectar computadoras incluyen bus, star, ring, y híbrida.

Topología Bus

Topología Bus requiere que todas las computadoras ó nodos, estén conectadas a un mismo cable. Algunas computadoras envían data, esa data entonces es transmitida (broadcast) a todos los nodos en la red. El termino BUS se refiere a un AUTOBUS actual que debe detenerse en cada parada en su ruta. Sólo el nodo al cual el mensaje fué destinado leerá el mensaje; el resto de los nodos lo ignoran.

Oficinas y negocios pequeños a menudo usaban redes del tipo bus, ya hoy día con el abaratamiento de los equipos de redes como son HUBS, Switches, etc, este no es el caso. Si este tipo de red crece normalmente evolucionan a tipo start. Redes del tipo BUS requieren un terminadores en cada punta para asegurar que el tráfico no podruzca eco en toda la red.

Las redes Bus son relativamente simple, económicos, fácil de operar, y confiable. Además es muy eficiente en su uso de los cables. Una desventaja es que se dificulta diagnosticar problemas debido a que es difícil apartarlos; si un cable se rompe, la red entera puede ser afectada. Durante los momentos de mucho uso la red se tornará lenta.

Topología Star (Estrella)

Topologías del tipo estrella conectan nodos de redes a través de un dispositivo central, normalmente un hub (más adelante explicaremos que son los hubs). Como cada conexión de computador termina en el hub, este diseño reduce el riesgo de fracaso total de la red. Por ejemplo en el caso de la ruptura de un cable, sólo el nodo conectado directamente a este será afectado. El resto de la red funcionaría perfectamente bien.

Administradores de redes pueden resolver problemas mucho más fácil en redes del tipo estrella ya que los problemas son aislados automáticamente por el diseño. La red no es afectada cuando se desarrolla problema en un nodo. Expandir la red es simple y automática, la administración y el monitoreo puede ser centralizado. La desventaja es que como todo es centralizado a través del hub, si este falla toda la red se caerá.

Topología Ring (Anillo)

Topologías de tipo Ring no tienen un punto central de conexión. Estas redes tienen un cable que conecta un nodo al otro hasta que se forme un anilla. Cuando un nodo envía un mensaje, el mensaje es procesado por cada computador en el anillo. Si la computadora no es el nodo de destino, esta pasará el mensaje al próximo nodo hasta que el mensaje llegue a su destino correcto. Si por alguna razón el mensaje no es aceptado por ningún nodo, este dará la vuelta completa retornando a quien envió el mensaje originalmente.

Las redes de tipo Ring a menudo se conectan a través de un dispositivo central llamado Unidad de Acceso de Multiple Estaciones (Multistation Access Unit, MSAU), más adelante explicaremos que son estas unidades. Aislar los problemas en esta topología es difícil. Si se caé un nodo se caé todo la red.

Una ventaja de usar topología ring es que todas las computadoras tienen el mismo acceso a toda la data. Durante los período de uso pico, el rendimiento es igual para todos los nodos. Las redes tipo ring también ejecutan excelente en redes de alto tráfico de información. Una desventaja es que la expansión de la red o su reconfiguración afectará la red por completo.

Redes Híbridas

Puede ser que confronte grandes redes que combinen las topologías bus, star, y ring. Esta combinación permite expansión hasta en redes del modelo empresarial (enterprise networks). Por ejemplo puede ser que confronte una red star y una star bus. En una red star ring, dos ó más topologías star son conectadas usando un MSAU como un hub central.

En una red star bus, dos o más topologías star están conectadas utilizando un bus trunk. El bus trunk sirve como un backbone de la red.

Observe que cada red star contiene dos nodos y esta conectada por bus trunks. Esta topología es excelente para compañías grandes porque el backbone puede implementar media que soporte transferencias alta de data. Una ventaja de usar redes Híbridas es que las expansiones son relativamente simple. La redes no son normalmente afectadas si uno de los nodos falla. Pero si el Hub falla, las computadoras conectadas a ese hub no podrán comunicarse. Las conexione entre el hub que ha fallado y los otros también estarán fallida.

Topología Mesh

Topología Mesh (Malla) interconectan dispositivos con multiple rutas para asegurar que exista redundancia. Todos los dispositivos están interconectados para así asegurar que la ruta más conveniente entre un dispositivo y otro es tomada en cada momento.

En una topología mesh, si una conexión es terminada ó interrumpida, otra conexión puede ser elegida para transferir la data a su nodo de destino. Una desventaja es que dispositivos (hardware) adicional tiene que ser empleado lo cual hace que que la topología mesh sea más costosa.

Sistemas Operativos de Redes

Un Sistema Operativo de Redes (Network Operating System, NOS) administra recursos en una red. Sus funciones incluye administrar multiple usuarios en una red, proveyendo acceso a los archivos y servidores de impresión, y implementación de la seguridad de la red. GNU/Linux es uno de los mejores y más populares, al día de hoy, NOS.

Un NOS le permite a los clientes acceso remoto a los dispositivos, como si fuesen parte del equipo local del cliente. Ellos te permiten procesar peticiones desde los clientes y decidir si el cliente puede usar un recurso en particular.

Al igual que la relación cliente/servidor, una parte del NOS debe encontrarse ejecutando en el cliente, y la otra parte debe ejecutarse en el servidor. En una red peer-to-peer, cada cliente puede operar como ambos cliente ó servidor.

Muchos NOSs pueden operar entre ellos, esta característica es conocida como interoperabilidad. Esta característica facilita que las corporaciones pueda crear redes aunque sus clientes y servidores se encuentren ejecutando diferentes sistemas operativos. En la mayoría de los casos, software deberá ser instalado en los servidores y clientes para que exista esta interoperatividad.

Servicios Básicos de Red

En un ambiente de redes hay disponible varios facilitadores básico:

Distributed File System (Dfs)

Sistemas de Archivos Distribuidos permiten a los clientes en una red accesar una jerarquía de archivos en un servidor central. Clientes y Servidores UNiX y GNU/Linux por lo general utilizan Network File System (NFS) para proveer Servicios de Dfs, mientras que MicroSoft utiliza el protocolo Server Message Block (SMB). Sistemas de Archivos Distribuidos son generalmente especificos a un sistema operativo de red cliente, aunque existen un número de utilitarios para accesar a NFS desde Windows y a SMB desde UNiX y GNU/Linux.

File transfer

Transferencia de Archivos permite que archivos individuales sean copiados desde un sistema a otro. Utilitarios de este proceso van desde mecanismos simple como el proceso único de Kermit al protocolo multitarea sofisticado proveído por el TCP/IP File Transfer Protocol (FTP).

Remote printing

Impresión Remota le permite al usuario en un siste imprimir un archivo en una impresora de otro sistema en la red. El otro equipo pueda o no estar ejecutando el mismo sistema operativo ó no. Esto sucede normalmente transparente al usuario y no requiere ninguna acción especializado por el usuario o el programa que este utilizando para generar el archivo a imprimir.

Electronic mail

Correo Electrónico permite que diferente usuario en sistemas diferentes se envíen mensajes entre ellos. Los sistemas de correo electrónicos son cada día más y más sofisticados; no es poco común poder enviar mensajes de voz y otros tipos de documentos multimedia con este método de comunicación.

Telnet

Protocolo de Terminales Virtuales proveé conexión entre máquinas y así permitiendo acceso remoto.

Hypertext Transfer Protocol (HTTP)

Protocolo de Transferencia de Hipertexto acepta peticiones de para documentos de del formato HTML (Hypertext Markup Language) y transfiere estos documentos al navegador (browser) remoto. Peticiones subsecuentes pueden incluir gráficos embebidos, fuentes tipográficas, javascripts y applets de java.

HISTORIA DEL INTERNET

A través de su historia, donde se originó, quien la controla y quien la administra podemos poner en perspectiva esta tecnología tan reciente que a veces tiende a ser confusa para muchos de sus usuarios. En esta sección, discutiremos los siguientes tópicos:

- Interoperabilidad TCP/IP
- Internetworking y la Red Corporativas
- Evolución del Internet
- Puntos de accesos de la Red/NetworkAccess Points (NAPs)
- Las Autoridades Relacionadas con el Internet
- El RFC (Request for Comments)

TCP/IP y la Interoperabilidad

El protocolo de red TCP/IP es el más usado hoy en día, esto no fué siempre el caso. Anteriormente, ya no muy reciente, redes de Novell usaban IPX/SPX (Internetwork Packet Exchange/ Sequenced Packet Exchange) como su protocolo por defecto de redes. Muchas redes de Novell aún continúan utilizando IPX/SPX y siguen siendo productivas. Muchas redes aún siguen usando protocolos que no son TCP/IP; estos protocolos son tan productivos como cualquier otro. Pero, si una red utiliza un protocol como por ejemplo NetBIOS (Extended User Inretface, NetBEUI) y otra usa el IPX/SPX, ellas no se podrán comunicar entre si. Estas redes deberán emplear dispositivos especiales, llamados gateways entre ellas para traducir los diferentes protocolos, pero una solución tal vez un poco más efectiva fuese adoptar el protocolo de red TCP/IP.

De echo estas redes no necesitan abandonar por completo los protocolos que ha estado usando tradicionalmente. Ellas pueden usar un protocolo internamente y usar TCP/IP como el protocolo que llevará acabo el transporte de la información entre su red y las otras.

El TCP/IP puede permitir que diferentes tipo de redes se comuniquen entre ellas. Usando sólo un router, el TCP/IP le permite a su LAN ó WAN existente intercomunicarse con otras.

El TCP/IP trabaja con muchas topologías diferentes. También puede funcionar en paralelo con otros protocolos a través de una sólo tarjeta de red (Network Interfase Card, NIC), ó encapsular otros protocolos dentro de los paquetes TCP/IP para interconectar dos redes que no usan TCP/IP vía una red de TCP/IP como es el Internet. Así esta actúa como un puente ideal que permite que redes LAN y WAN actuar como backbones para una empresa.

Internetworking y la Red Corporativa

Hemos visto que el protocolo TCP/IP es ideal para internetworking (trabajo a través del Internet) ya que este permite que diferentes sistemas trabajan unidos. Este tipo de capacidad de trabajo de interplataformas significa que un equipo tradicional, digamos de IBM, puede comunicarse con soluciones cliente/servidores de hoy como lo son UNiX, Windows NT, Macintosh, y redes Novell. Los mainframe tradicionales anteriores pueden entonces trabajar entre ellas también. Como el protocolo TCP/IP es independiente de los fabricantes, este permite que trabajadores conectados vía el Internet se conecten desde cada sistema sin sacrificar la fortaleza de sus sistemas operativos ó su método de redes.

El protocolo TCP/IP ha sido atractivo para internetworking porque permite que las corporaciones y redes que sigan utilizando sus pasadas inversiones por mucho más tiempo. Por esto es que aunque el Internet y el

internetworking son revolucionarios, este protocolo presenta a los negocios una atractiva alternativa para que no tengan que eliminar flotillas de sistemas por anticuados. Con una buena planificación y trabajo, las compañías pueden hacer que sus equipos viejos se comuniquen con los equipos y sistemas nuevos y que trabaje con cualquier sistema sobre el Internet.

Evolución del Internet

El Internet se formó en el año 1968 cuando el Departamento de Defensa de los Estados Unidos creó la Agencia para Proyectos de Investigación Avanzada (Advanced Research Projects Agency, ARPA). Su primer red computacional global, fué el ARPANET (Red de la Agencia para Proyectos de Investigación Avanzada), que le permitió al gobierno y a los investigadores trabajar desde cualquier localidad que se pudiese conectar a la red. Las características de diseño del ARPANET contemplaba multiple clientes (hosts) y múltiples conexiones entre los clientes, lo cual reduciá significativamente los chances de fracasos total de la red. No existía un hub central, lo cual hubiese creado un punto de vulnerabilidad; a diferencia, el control estaba distribuido en toda la red. Esta decentralización resulto en una red robusta y confiable que continuaría funcionando aunque muchos de sus clientes se cayesen y resultasen incapacitados.

Ya en el año 1971, el ARPANET consistía de 23 computadoras a todo lo ancho de los Estados Unidos, y al 1972, creció a 40. El año siguiente se extendió a Europa. Fué en este momento que esta colección de computadoras conectada en una red fué conocida como el Internet.

Al principio de los 80s, la versión del sistema operativo UNiX de Berkeley ya soportaba el protocolo TCP/IP, y en el 1981, TCP/IP pasó a ser el estándar oficial del Internet. El primero de Enero del 1983, el TCP/IP fué adoptado como el protocolo oficial del Internet. También en las década de los 80, el Departamento de Defensa asignó el proyecto ARPA a la Fundación Nacional de Ciencias (NSF). La NSF es una agencia independiente del gobierno de los Estados Unidos que promueve los avances de las ciencias y la ingeniería. La NSF incrementó el número de supercomputadores a cinco y agregó acceso a más redes, expandiendo sitios a negocios, universidades, al gobierno y instalaciones militares. Estos centros fueron conectados a través de líneas telefónicas de 56-Kbps que creaban redes regionales, con cada centro con un supercomputador sirviendo de hub para conexión de una región en particular.

El tráfico de red se incrementó significativamente. Un contrato fué obsequiado a la compañía Merit Network, Inc., lo cual permitió que las conexiones entre varias localidades operaran a velocidades de 1.5 Mbps.

En los años subsecuentes, se incluyeron más y más compañías privadas al Internet, hoy en día tecnologías existen para que se puedan adquirir velocidades de más de 2 Gbps y todos los días se establecen nuevos records de velocidad de transferencias. En la actualidad, tanto el hardware y los enlaces de comunicación que se requieren para interconectar el Internet son financiadas por una combinación del sector privado y del estado ó gobierno.

Punto de Acceso de la Red (Network Access Points, NAP)

Ya hemos dicho que el Internet es una serie de redes interconectadas. Un NAP es un punto donde se unen una red de alta velocidad y otra. Los tres puntos NAP claves en los Estados Unidos se encuentran en New York, Chicago, y San Francisco. Estos tres NAP son ejecutados por compañías telefónicas. Estos tres puntos de intercambio más el de Washington, D.C., son los cuatro originales en los Estados Unidos.

Estas redes de alta velocidad son las llamadas los backbones del Internet, ya que ellas proveen la conectividad esencial para el resto del Internet. Los Backbones pueden cubrir larga y cortas distancias, y redes más pequeñas típicamente se conectan a ellas.

La red backbone conectada por un NAP es una Red de Servicios Backbone de muy alta velocidad (very high-speed Backbone Network Service, vBNS). La mayoría de los NAP en uso al día de hoy son de velocidad en exceso de 1 Gbps y están diseñadas para reducir la congestión debido al uso incrementado continúo del Internet.

Por una mayor parte de la vida del Internet, agencias regionales y gubernamentales cargaban con la responsabilidad para proveer la conexión física. Empezando en el año 1995, los Proveedores de Servicios de Internet comerciales (Internet Service Providers, ISP) empezaron a financiar el Internet. Este grupo de ISP es llamado la Red Nacional de Investigación y Educación (National Research and Education Network, NREN). Esta usa los NAP para la conectividad.

Segmentos/Segments

Un segmento es un pedazo o parte de una estructura mucho más grande. En el Internet, un segmento puede ser parte del backbone que conecta San Francisco a Chicago. A escala menor, un segmento puede ser la conexión de la red en su oficina a su NAP. El término segmento también se refiere a una subred en una LAN.

Las Autoridades Relacionadas con el Internet

La autoridad de Internet descanza sobre el organismo llamado Sociedad del Internet (Internet Society, ISOC). ISOC es una organización de membrecía voluntaria de la cual sus objetivos es promover el intercambio de información a través del uso de la tecnología del Internet.

La ISOC escoge voluntarios quienes son responsables de la administración técnica y la dirección del Internet; estos voluntarios son denominados la Junta de Arquitectura del Internet (Internet Architecture Board, IAB). Usted puede visitar la sociedad del Internet en el sitio web www.isoc.org.

Otra organización voluntaria, llamada Equipo de Tareas de Ingeniería del Internet (Internet Engineering Task Force, IETF), se reúne regularmente para discutir problemas de operacionales y técnicos del Internet. Las recomendaciones son hechas vía grupos de trabajo dentro de la IETF y pueden ser enviados a la IAB para ser declarados como Estándares del Internet. El presidente de la IETF y los administradores de áreas conforman el Grupo Conductor de Ingenieros del Internet (Internet Engineering Steering Group, IESG).

Otra organización, llamada el equipo de Tareas de Investigación del Internet (Internet Research Task Force, IRTF), es responsable de investigación y el desarrollo de nuevas tecnologías aplicadas a la redes. El Grupo Conductor de Investigación del Internet (Internet Research Steering Group, IRSG) establece prioridades y coordina actividades de investigación.

Petición de Comentarios (Request for Comments, RFC)

Los RFC son documentos publicados de interés para la comunidad del Internet. Estos incluyen información detallada acerca de protocolos de Internet estandarizados y aquellos que se encuentran aún varios estados de su desarrollo. también incluyen documentos informativos relacionados con Estándares de protocolos, asignación de numerosa (E.j.,., números de puerto), requerimientos de host (E.j.,., Enlace de Data, Red, Transporte, y capas de Aplicación OSI), y requerimientos de enrutadores.

Los RFC son identificados por números. Mientras más alto es el número, es una indicación directa que es más actualizado el RFC. Asegurece de estar leyendo el más actualizado o reciente de los RFC durante sus investigaciones. El sitio web recomendado como referencia de los RFC es el sitio web www.rfc-editor.org/rfc.html. Observe que si un RFC ha sido actualizado, su listado nombrará su número de RFC reemplazo.

Estados de los Protocolos

Antes de que un protocolo se establezca como un estándar, este pasa por varios estados de madurez: experimental, propuesto, prueba, y estándar. Si un protocolo se torna obsoleto, se clasifica como histórico. Para que un protocolo puede progresar a través de estos pasos, el protocolo debe ser recomendado por el IESG del IETF.

Aquí le presentamos los diferentes estados de madurez de los RFC:

- **Experimental**
  Estos protocolos deben ser usados en situaciones de ejercicios y pruebas de investigación. Su intención no es para sistemas bajo operación normal sino para sistemas participando en la investigación del experimento.
- **Propuesto**
  Protocolos que están siendo considerados para una futura estandarización. Su puesta en prueba e investigación son promovidas (9999encouraged), es preferible que esas investigaciones sean llevadas acabo por más de un grupo. Estos protocolos serán revisados antes de pasar a la próxima etapa de su desarrollo.
- **Draft/Prueba**
  Estos son protocolos que están siendo seriamente considerados por el IESG para ser promocionados al estado de estándar del Internet. Pruebas son altamente 9999encouraged, los resultados de las pruebas son analizadas, y se requiere retroalimentación de información. Toda información pertinente a esta debe ser enviada al IESG. Muy a menudo cambios son efectuados en esta etapa de prueba; el protocolo dependiendo de estas propuesta es colocado en la etapa correspondiente a las propuesta.
- **Estándar**
  Estos protocolos son determinados por el IESG para convertirse oficialmente en protocolos Estándares del Internet. Los protocolos estándar se categorizan en dos tipos: esos que se aplican a todo el Internet y aquellos que sólo aplican a ciertas redes.

Otros estados de protocolos incluyen:

- **histórico**
  Estos son protocolos que han sido reemplazados por versiónes más recientes ó que nunca recibieron suficiente apoyo e interés para su desarrollo. Es muy difícil que un protocolo histórico se convierta en un protocolo estándar del Internet.
- **Informativo**
  Estos protocolos fueron desarrollados fuera del IETF/IESG (E.j.,. protocolos desarrollados por los fabricantes u otras organizaciones de estandarización). Estos protocolos son publicados para el beneficio de la comunidad del Internet.

Estándares del Internet

Un protocolo ó conjunto que ha sido elevado a estándar es indexado como estándar STD, como es el STD 5. Todos los protocolos, hasta los Estándares, son indexados como RFC porque los RFC nunca son eliminados, estos sólo cambian de estado del protocolo. Por ejemplo, IP es STD 7 Así como RFC 793. En algunos casos varios RFC pueden convertirse en un sólo STD. Por ejemplo, IP, ICMP, y IGMP son indexados todos en un sólo STD 5 aunque existen tres RFC diferentes: 791, 792, y 1112, respectivamente. estudiaremos estos protocolos en la próxima sección.

RFC de Referencia

Los siguientes son RFC de referencia que son importante y usted debe estar familiarizado con cada uno de ellos:

- **estándar Oficial del Protocolo de Internet, RFC 2700, STD 1**
  Este RFC lista el estándar actual del Internet Así como el estado actual de todos los RFC.
- **Números Asignados, RFC 1700**
  Lista el estado actual de los parámetros, como son los nombres y palabras claves, usadas en el Internet. Este incluye los números de protocolos asignados a los protocolos de Internet. Por ejemplo, el protocolo IP está representado por el número decimal 4. También se incluyen asignaciones bien reconocidas y registradas de puertos. Aprenderás sobre la asignación de números a través de este curso.
- **Requerimientos de Hosts de Internet, RFC 1122 y 1123**
  Este par de RFC define los requerimientos de software de host de Internet. Ellos definen los requerimientos únicos de protocolos dentro de la arquitectura del Internet y lista las características e implementación detalladas del protocolo (E.j.,., las especificaciones de los protocolos se identifican como ***must***, ***must not***, ***should***, ***should not***, y ***may***).

- **Requerimientos de Enrutadores de IP Versión 4, RFC 1812**
  El RFC 1812 define los requerimientos únicos de los enrutadores de IPv4 de Internet. Este actualiza el ahora histórico RFC 1716. Los requerimientos del enrutador incluyen la tecnología actual de enrutar.
- **Requerimientos de Enrutadores de IP Versión 6, RFC XXXXX**
  El RFC XXXXX define los requerimientos únicos de los enrutadores de IPv4 de Internet. Este actualiza el ahora histórico RFC 1716. Los requerimientos del enrutador incluyen la tecnología actual de enrutar.

Ejercicios 1-1: Ubique los Documentos de los RFC

En este ejercicio, se le pide que ubique en el Internet los sitios de donde puede descargar los RFC. Usted debe investigar más de un sitio web ya que no todos los portales actualizan sus RFC con regular frecuencia. Soluciones para este ejercicio se proveen en el Apéndice B.

1. Desde el buscador www.google.com en un browser ubique tres sitios web que mantienen el listado de RFC. Liste los sitios por su URL como repuesta.

2.- Elija un sitio RFC y use el browser para ubicar los RFC 1250, 2068, y 1441. Liste los títulos de cada uno de ellos. Determine si el RFC es la última versión. Si no lo es, identifique cual es el más reciente. Usted deberá usar un editor de RFC que permita peticiones, como el que se encuentra en el sitio web www.rfc-editor.org/rfc.html. Elija buscar/Search desde el enlace índice de RFC que le permite buscar por número, titulo y autor.

RFC #	Titulo del RFC	RFC más actual.
RFC1250		
RFC 2068		
RFC 1441		

3. Responda estas preguntas basada en la información en los RFC 2600 y 1700.
   a. ¿Es OSPFv2 un protocolo estándar del Internet? ¿Es RIP-2 un estándar?
   b. ¿En que año se constituyó POP3 como un protocolo estándar del Internet? ¿Y el TCP? ¿Y el IP?
   c. ¿Cúal es el STD para el protocolo de servicios NetBIOS?
   d. ¿Cúal es el rango de los números de puertos más reconocidos (well-known)?
   e. ¿Para que es el puerto 110 usado?
   f. ¿Cúal es el rango de los números de puertos registrados?
   g. ¿Para que es el puerto 533 usado?

EL MODELO DE COMUNICACION DE DATA

En esta sección introducimos el modelo de referencia para la Interconexión de Sistemas Abiertos (Open Systems Interconnection, OSI) y el modelo de arquitectura de Internet. Estos modelos sirven para definir el mecanismo de todo protocolo de redes del TCP/IP usado para la comunicación de data.

En esta sección, discutiremos los siguientes tópicos:
- Protocolos de Comunicación
- Modelo de Referencia OSI
- Separación de Funcionalidad
- Arquitectura de Internet
- Paquetes/Packets

- Protocolos de Aplicación, Transporte, y Red
- Multiplexado/Demultiplexing
- Protocolos de Interfase Serial Especializada

Protocolos de Comunicación

Existe una necesidad de cierta forma de estandarización entre entidades que se comunican. Un protocolo es nada más que un conjunto de reglas y convenciones que describen como dos entidades se comunican una con la otra. Un ejemplo simple es un protocolo de comunicación es el lenguaje humano natural, digamos el español ó el ingles, estos permiten a humanos intercomunicarse uno con el otro y entender entre si que se esta diciendo. Es obvio que cuando dos computadores se intercomunican, el protocolo que se emplea deberá ser más definido, pero la idea básica es la misma.

Si dos personas no hablan un mismo idioma, ellos no podrán entenderse. Lo mismo pasa con dos computadoras que no entiende sus diferentes protocolos, para entenderse deberán emplear el mismo protocolo de comunicación para poder llevar acabo con éxito la comunicación.

Capas de Protocolos (Layers)

Los protocolos usados para intercomunicar computadores son bien complejos. Para orden administrar esta complejidad, los protocolos son divididos en una serie de capas, cada una de la cual es responsable de manejar cierto aspecto de la comunicación.

Los protocolos de cada capa se comunican con protocolos de su misma capa ó nivel en el sistema remoto. Realmente la data se transmite en orden decendiente a través de las diferentes capas en el sistema que envía ascendente en la capa del sistema que recibe. En cada capa el sistema que envía, coloca información de control al mensaje original. En el sistema receptor, esta información de control es interpretada por el protocolo homologo (peer) y esta información es removida del mensaje hasta Así llegar hasta el equipo receptor intacto como se envió el mensaje original.

Una colección de protocolos estructurado de esta manera aquí descripta es conocido como una pila (protocol stack).

Información de Control

Información de control agregada a un mensaje original al pasar entre protocolos puede ser comparada a un documento que ha sido colocada en una serie de sobres más grandes y más grande cada vez. Cada sobre tiene una dirección colocada en el para que Así el protocolo en ese mismo nivel en el host que recibe pueda entender para quien esta dirigida la data.

Al momento de recibir la data en el otro extremo, cada capa procede a destapar su sobre y pasa hacia arriba en la pila de protocolos el sobre que yace dentro del que esta dirigido a el, el determina a quien va dirigido por la dirección colocado en el sobre inmediatamente después del de el.

Cada capa en l apila no sabe nada de lo que contiene el sobre que el recibe desde arriba de la pila o por debajo de ella. Esto mantiene un diseño limpio y separado que permite a las capas trabajar independientemente.

El Modelo de Referencia OSI

El Modelo de Referencia OSI fué definido por la Organización Internacional para la estandarización (International Organization for Standarization, ISO). Fué introducida en el año 1983, el modelo OSI tiene tres funciones prácticas:

- Proveer a los desarrolladores los conceptos necesarios y universales para que puedan desarrollar y perfeccionar los protocolos.
- Explicar la estructura de trabajo usada para conectar sistemas heterogéneos. Más simple, permite a los clientes y servidores comunicarse aunque este usando aplicaciones y sistemas operativos distintos; todo lo que ellos necesitan es un protocolo común entre ellos como son por ejemplo el TCP/IP ó el IPX/SPX.
- Describir el proceso de la creación de los paquetes. Discutiremos más sobre la creación de los paquetes más adelante.

Redes son diseñadas y construidas usando el modelo OSI, en la misma manera que edificaciones son construidas usando planos. Por ejemplo, Novell NetWare, Windows NT, y GNU/Linux (UNiX) están todos basados en el modelo OSI. Este misma base arquitectónica de trabajo es que permite que estos sistemas se puedan intercomunicar.

Además, cuando quiera se discute de protocolos, como por ejemplo el IP y el IPX, normalmente estos son enlazados a su capa OSI. En este ejemplo, ambos protocolos se encuentran en la capa OSI de Red. El modelo OSI provee los conceptos y la nomenclatura que usted necesita para poder discutir lo que es la creación de paquetes y los protocolos de redes.

Como se Comunican las Capas

El modelo OSI describe la interacción entre las capas individuales Así como entre los hosts en la red. Un ejemplo de Cliente/Servidor será usado para explicar como trabaja típicamente el modelo OSI. La

Capa	No.	Descripción
Aplicación	7	Esta es la capa presentada al usuario atraves de las aplicaciones disponibles en el ambiente OSI de su sistema operativo; esta capa soporta transferencia de archivos, administración de redes, entre otros servivios.
Presentación	6	Capa responsable de proveer transformaciones útiles sobre data para soportar interfases de aplicaciones estandarizadas y servivios de comunicación general. Por ejemplo, convierte texto de (ASCII) a Binario Extendido (EBCDIC).
Sesión	5	Establece, administra y termina conexiones (sesiones) entre las aplicaciones que colaboran entre ellas. Esta capa además agrega información de tráfico.
Transporte	4	Proveé transporte confiable y transparente entre dos puntos (Ej., los hosts de origen y destino). También proveé soporte de recuperación de errores y control de tráfico. Protocolos de estado de conxiones residen en esta capa.
Red	3	Esta capa es responsable de reenviar y enrutar los datagramas, (Protocolos sin estado de conexión residen en esta capa.
Enlace de Data	2	Proveé transferencia de data confiable atraves del enlace físico. Las tramas son transmitidas con la sincronización necesaria, control de error, y el control de flujol. Simplificando, esta prepara la información para que pueda ser enviada al cable físico. En las series de estandares IEEE 802 de de las LAN (un grupo de estandares populares de redes que estudiaremos más adelante), la capa de Enlace de Data está dividida en dos subcapas, el Control de Enlace Logico (Logical Link Control, LLC) y el Control de Acceso del Medio (Media Access Control, MAC). La capa LLC es responsable del control de error y flujo, y la capa de MAC es responsable de colocar la data en el cable.
Fisica	1	Capa que se concierne con lo relacionado con la transmisión de flujo de bits no estructurado sobre un enlace físico. Es responsable por las características mecanicas, electricas y procedural para establecer, mantener y deactivar el enlace físico.

columna izquierda contiene las siete capas del modelo OSI que existen en el cliente. A la derecha la colum-

na con las siete capas que existe en el Servidor.

Cliente		**Servidor**
Aplicación	**<<<------>>>**	**Aplicación**
Presentación	**<<<------>>>**	**Presentación**
Sesión	**<<<------>>>**	**Sesión**
Transporte	**<<<------>>>**	**Transporte**
Red	**<<<------>>>**	**Red**
Enlace de Data	**<<<------>>>**	**Enlace de Data**
Física	**<<<------>>>**	**Física**

Si el cliente realiza una petición al servidor, la petición puede ser que empiece con un click del mouse por un usuario navegando en una página web (Capa de Aplicación). La petición viaja hacia abajo en la estructura del modelo OSI hasta que este llegue a la capa de Enlace de Data donde esta es entonces colocada en el alambre, al cable, ó cualquier otro medio que es usado en su red (la capa Física).

La petición del cliente viaja a través del cable hasta que esta llega al servidor. La capa de Enlace de Data toma la petición del cable (capa Física) y la envía hacia arriba en la estructura del modelo OSI. Cuando la petición llega a la capa de Aplicación del servidor, la petición es procesada. El servidor entonces retorna una repuesta al cliente, la cual puede ser una página Web nueva, usando el mismo método.

En la redes, la información como las peticiones del cliente y las respuestas del servidor son enviados a través de la red vía paquetes. Los paquetes sean discutidos más adelante en esta sección.

Separación de Funcionalidad

La motivación principal detrás del Modelo de Referencia OSI es la separación de funcionalidad entre las capas. Cada capa esta diseñada para ejecutar una función en específico y servir de interfase a otras capas en una manera bien definida. Los detalles internos de una capa no deben afectar la operación de las capas inferior o la superior a ella en la pila. De esta manera una capa puede ser removida completamente y reemplazada con otra. Siempre y cuando la interfase entre dos capas sea consistente, entonces este cambio no debe afectar la funcionalidad total de la red.

Esta forma idealista no siempre se puede lograr, pero la idea básica puede ser vista en un sin número de sitios. Una área de esta es la de direccionar el sistema. La capa de red superior debe operar independientemente del medio de red usado; por ejemplo, el uso de Ethernet ó Token Ring no debe afectar los protocolos de alto nivel. Esto nos deja dicho que los sistemas deben direccionar en una manera independiente del medio de conexión. Pero la capa de Enlace de Data debe poder direccionar sistemas, y el formato y la interpretación del direccionamiento dependerá del medio de conexión.

Así que la capa de Enlace de Data utiliza su propio formato de direccionar, el cual es entonces mapeado al nivel superior, direcciones independiente de la red como parte de la interfase entre las dos capas. Si un medio diferente de conexión es usado, entonces se requerirá de la interfase de ejecutar una forma diferente del mapeado de direcciones apropiado para la nueva capa de Enlace de Data. Pero, la operación del protocolo de la capa de Red no debe ser afectada por este cambio.

Arquitectura de Internet

Al igual que otros modelos de redes, la arquitectura de Internet divide los protocolos en capas. Cada capa es responsable de tareas de comunicación en específico. La arquitectura de Internet consiste de cuatro capas, cada una coincide con las capas del Modelo de Referencia OSI. Recuerde que existen varios

modelos de arquitectura de Internet, cada uno un poco diferente al otro. Aquí sólo trabajaremos con la versión de cuatro capas.

- **Capa de Aplicación**
- **Capa de Transporte**
- **Capa de Internet**
- **Capa de Acceso a la Red**

La siguiente tabla muestra la arquitectura de Internet y su equivalente en el modelo de capas OSI.

La Capa de Acceso a la Red

La capa de Acceso a la Red corresponde a las capas física y Enlace de Data del modelo OSI. Esta capa acepta paquetes de capas más alta y los transmite a través de la la red de conexión, manejando todos

Modelo de Referencia OSI	Equivalente Modelo de Arquitectura de Internet
Aplicación	Aplicación
Presentación	
Sesión	Transporte
Transporte	
Red	Internet
Enlace de Data	Acceso a la Red
Física	

los detalles del medio de las interfases de hardware con la red. Esta capa normalmente consiste de:

- Los manejadores de los dispositivos del sistema operativo
- La tarjeta de la interfase correspondiente
- Las conexiones físicas

Para las LAN basadas en tecnologías Ethernet, la data enviada por la media es referida como tramas de Ethernet, las cuales varían en tamaño desde 64 hasta 1,518 bytes (1,514 bytes sin la revisión de Redundancia Cíclica).

La Capa Internet

La Capa de Internet corresponde a la capa de Red del modelo OSI. Es responsable de direccionar y enrutar los paquetes en las redes TCP/IP. Un paquete recibido desde la capa de Transporte es encapsulado en un paquete IP. Basado en la información del host de destino, la capa de Internet usa un algoritmo de enrutamiento para determinar si envía el paquete localmente ó si necesita enviarlo a una pasarela (gateway) por defecto.

Los protocolos usados en la capa de Internet son:

- Protocolo de Internet (IP)
- Protocolo de Control de Mensajes de Internet (ICMP)
- Protocolo de Administración de Grupos de Internet (IGMP)
- Protocolo de Resolución de Direcciones (ARP)
- Protocolo de Resolución de Direcciones Inversas (RARP)

Capa de Transporte

La capa de Transporte del modelo de Arqitectura de Internet corresponde a las capas de Transporte y la de Sesión del modelo OSI. La capa de Transporte acepta la data desde la capa de Aplicación y le provee el flujo de información entre los dos hosts. Estos dos protocolos que siguen se encuentran en la capa

de Transporte:

- Protocolo de Control de Transmisión (TCP)
- Protocolo de Datagramas de Usuarios (UDP)

La capa de Transporte también divide la data recibida desde la capa de Aplicación en trozos de data más pequeños, llamados paquetes, antes de pasarla a la capa de Internet. La capa de Transporte también es conocido como la capa Host-a-Host, la capa Punta-a-Punta, ó la capa Fuente-a-Destino.

La Capa de Aplicación

La capa de Aplicación de la Arquitectura de Internet corresponde a las capas de Presentación y la de Aplicación del Modelo OSI. La capa de Aplicación interactúa con el protocolo de la capa de Transporte para enviar ó recibir data.

Los usuarios pueden invocar programas de aplicaciones, como es el protocolo de terminal remoto (Telnet), Protocolo de Transferencia de Archivos (FTP), Protocolo de Transferencia de Correo Simple (SMTP), ó Protocolo de Administración de Redes Simple (SNMP) para accesar los nodos en el Internet. La capa de Aplicación también suele ser referida de nombre la capa de los Procesos.

Los Paquetes

Un paquete es un pedazo fijo de información enviado a través de la red. En el momento que enviamos información a través de cualquier red, empezamos el proceso de creación de paquetes. Un paquete consiste de tres elementos: un cabezal (header), la data en si, y un finalizador (trailer). A menudo los paquetes también son referidos como datagram ó frame. Aunque este uso es aceptable la mayoría de las veces, un paquete es un nombre genérico usado para referirse a cualquier pedazo de información pasada por una red. Un datagram es un paquete en la capa de Red del modelo OSI. Un frame (trama) es un paquete en la capa de Enlace de Data . Aunque esta diferencia existe ellos son intercambiados sinónimamente.

El header contiene información variada, como es información referente a direccionamiento ó señales de alerta a la computadora entrante. El paquete contiene la data original, como por ejemplo un correo. El trailer normalmente contiene información que valida el paquete. Por ejemplo, puede contener la información Revisado de Redundancia Cíclica (CRC).

Cyclic Redundancy Check (CRC)

Un CRC es un calculo matemático que le permite al computador que recibe validar si un paquete es valido. Cuando un host envía un paquete, este calcula un CRC, entonces procede a agregar esta información al trailer. Cuando el host que recibe lee el paquete, este genera su propio CRC, entonces lo compara con el CRC que esta almacenado en el trailer. Si igualan, el paquete no esta dañado u el host que recibe procesa el paquete. Si los CRC no igualan, el host recibiendo lo descarta por completo.

Creación de Paquetes- Agregando los Headers

La creación del paquete empieza en la capa 7 del modelo OSI (capa de Aplicación) y continúa en todo el proceso hasta la capa 1 (capa Física). Por ejemplo, al enviar un correo ó transferir un archivo desde una computadora a otra, este mensaje ó archivo es sometido a una trasformación de un archivo discreto ó completo es convertido en muchos pedazos pequeños llamados paquetes. Empezando desde la capa de Aplicación del modelo OSI, el archivo continúa siendo dividido en pequeños pedazos hasta que el archivo discreto original sea, sea convertido en piezas suficientemente pequeñas, y más manejable pedacitos de información y entonces serán enviados a la capa Física.Cada capa agrega su propia información, llamada header, al paquete. Esta información permite que cada capa se comunique con las otras, y también permite que el computador recibiendo pueda procesar el mensaje.

Creación de Paquetes- Remover el Header

Ya hemos visto como el host que envía crea los paquetes. Cuando el host que recibe procesa un paquete, este efectúa lo contrario que el host que lo creo hizo y remueve casa header, empezando con el de la capa 1 (capa Física) y termina con al capa 7. Todo lo que queda al final del proceso es la data original sin ningún tipo de alteración, la cual puede ser procesada por el host que recibe.

Protocolos de Aplicación, Transporte, y Red

Todos los protocolos discutidos en en este capítulo pertenecen a una de tres divisiones del modelo OSI. La primera categoría de protocolos abarca las capas del modelo OSI de Aplicación, Presentación, y Sesión. Estos se denominan como protocolos de la capa de Aplicación. Los protocolos de la capa de Transporte sólo aplican a la capa de Transporte; protocolos de redes abarcan las capas de Red, Enlace de Data y la física.

Protocolos de la capa de Aplicación

Los protocolos de la capa de Aplicación, también conocidos como protocolos de la capa alta, permiten que las aplicaciones interactúen entre si a través de la red. Los protocolos más comunes de la capa de Aplicación incluyen:

Simple Mail Transfer Protocol (SMTP)
Protocolo de Transferencia de Correo Simple es un suite de protocolo TCP/IP usado para enviar mensajes de e-mail desde y a un host.

Bootstrap Protocol (BootP)
Aunque aparentemente oscuro, el Protocolo de Bootstrap, el BootP es responsable de enviar paquetes TCP/IP dirigidos a la información de configuración del host.

File Transfer Protocol (FTP)
Protocolo de Transferencia de Archivos también es parte del conjunto de herramientas TCP/IP, el FTP es usado para transferir archivos de un host a otro.

Hypertext Transfer Protocol (HTTP)
El World Wide Web usa este conjunto o suite de protocolos TCP/IP para interconectar páginas Web.

Simple Network Management Protocol (SNMP)
Suite de protocolos TCP/IP que permite que administradores de red diagnostiquen y administren redes sin importar la arquitectura de esta.

Server Message Block (SMB) protocol
Protocolo usado en redes de Microsoft, este protocol permite que clientes trabajen con servidores, para accesar archivos y hacer peticiones de otros servicios. GNU/Linux puede interactúar con equipos usando el protocolo SMB utilizando el suite de Samba.

Network File System (NFS)
Protocolo que permite compartir archivos e impresoras en redes UNiX.

Protocolos de la Capa de Transporte

La capa de Transporte provee transferencia de data confiable. Entre los protocolos usados en esta capa se incluyen:

- **Transmission Control Protocol (TCP)**
  Forma parte del conjunto de protocolo de TCP/IP, TCP provee transferencias confiable y administra las sesiones.
- **Sequenced Packet Exchange (SPX)**
  Parte del conjunto de protocolo IPX/SPX, el SPX es similar al TCP en que es quien administra las sesiones de comunicación.
- **NetBEUI**
  Permite que distintas aplicaciones en diferentes computadoras usando NetBIOS se comuniquen entre ellas; es un protocolo no enrutable. El uso más frecuente de NetBEUI es en redes de sistemas Microsoft.

Protocolo de la Capa de Red

Los protocolos de Red proveén información de enrutamiento a los enrutadores y direcciones a los host.

Entre los protocolos de Red se incluyen:

- **Internet Protocol (IP)**
  Parte del conjunto TCP/IP, IP es el responsable de direccionamiento del host y de enrutar los paquetes en cualquier red ejecutando TCP/IP, incluyendo el Internet.
- **Internetwork Packet Exchange (IPX)**
  IPX provee servicios de direcionamiento para el suite de Novell IPX/SPX.
- **NetBEUI**
  Este permite que diferente aplicaciones en distintas computadoras usando NetBIOS se comuniquen una con otra; es un protocolo no enrutable.
- **Ethernet**
  Este protocolo LAN fué creado por Xerox, Digital Equipment Corporation, e Intel. Es la tecnología LAN más usada.

Estudiaremos más profundamente estos protocolos a través del libro.

Protocolos de Internet

HTTP	FTP	Telnet	TFTP	Gopher
SMTP	SNMP	DNS	BootP	DHCP
Capa de Aplicación				

ICMP	IP	IGMP
ARP		RARP
Capa de Internet		

TCP	UDP
Capa de Transporte	

Media
Capa de Acceso a la Red

Cada capa de la arquitectura de Internet involucra protocolos, y cada protocolo tiene su RFC asociado. Esta sección, describiremos brevemente protocolos comúnmente usados en cada capa del Internet. Estos protocolos serán discutidos más detalladamente más adelante. Cada protocolo está listado con sus respectivos RFC(s).

La Capa de Acceso a la Red

La capa de Acceso a la Red puede variar considerablemente, dependiente de que tipo de tecnología es responsable de colocar la data en el medio de la red y de captarla del medio. Algunos ejemplos:

LAN	Ethernet, Token Ring, y Fiber Distributed Data Interfase (FDDI)
WAN	Frame Relay, serial lines, y Asynchronous Transfer Mode (ATM)

La Capa de Internet

Aquí listamos los diferentes protocolos de la capa de Internet y sus respectivos RFC:

IP-RFC 791, STD 5

IP es el método básico de transferir data usado en todo el Internet. Es el responsable de el direccionamiento de IP y ejecuta la función de enrutar, el cual selecciona una ruta para enviar la data a la dirección IP de destino.

La data se envía en formato de paquetes, también conocidos como datagramas. Un paquete es autocontenido, totalmente independiente de otros paquetes, que no requiere supervisión y carga suficiente información para enrutar desde el host de origen hasta el host de destino.

IP define como los enrutadores procesan los paquetes, cuando se deben generar mensajes de error, y bajo que condiciones los paquetes serán descartados.

ICMP-RFC 792, STD 5

ICMP es el protocolo de diagnóstico del TCP/IP. ICMP está especificado en los RFCs 844, 1256, y 1788. Este permite que los hosts y gateways (pasarelas) reporten mensajes de error a través de los mensajes ICMP. Si ocurre un problema en una red TCP/IP, un mensaje ICMP será generado.

Internet Group Management Protocol (IGMP)-RFC 1112, STD 5

El IGMP es usado para multicasting. En multicasting, una fuente encia un mensaje a un grupo de subscriptores (multicast

groups). Para que la transferencia de un multicast sea exitosa, los miembros deben identificarse y los grupos que les interesa deben hacerlo a un enrutador local. El IGMP le permite a los usuarios pertenecer y mantener membrecías en grupos multicast.

ARP-RFC 826, STD 37

ARP traduce las direcciones de Internet a direcciones físicas, como son las direcciones físicas de 48-bit de Ethernet.

Por ejemplo, asumimos dos hosts en una red, nodo1 y nodo2. Nodo1 sabe la dirección IP de node2. Pero si el node1 desea enviar un paquete al nodo2, este debe saber la dirección física, ó el hardware, de la dirección del nodo2. Para poder resolver la dirección IP a la dirección del hardware, el ARP envía un broadcast local y Así obtiene la dirección de hardware.

Una vez la dirección a sido resuelta, el ARP almacena la información para las peticiones futuras en un cache. El cache del ARP permanece por un período de tiempo que varía dependiendo del sistema operativo.

RARP-RFC 903, STD 38

Como su nombre lo implica RARP ejecuta la función en reverso del ARP. Este utiliza la dirección de hardware del nodo para requerir una dirección IP. RARP es por lo general usado durante la inicialización de una estación de trabajo sin discos para obtener su una dirección IP.

Por ejemplo, cuando un estación de trabajo se inicializa, el RARP leé la dirección de hardware única y envía un broadcast (difusión) sobre la red para pedir una dirección. Un servidor RARP responde a las peticiones y provee una dirección IP.

La Capa de Transporte

Aquí le presentamos los protocolos de la capa de Transporte y sus RFC que lo define a ellos:

TCP-RFC 793, STD 7

El TCP provee la administración de sesión entre los sistemas fuentes y destino. Este asegura que la data sea transportado en secuencia y que no se envíe data duplicada. El TCP es usado con aplicaciones que se comunican, estableciendo la sesión antes de transferir data, como son el FTP y el Telnet.

UDP-RPC 768, STD 6

El protocolo UDP provee una comunicación simple de paquete. Un paquete UDP es creado por cada operación de las aplicaciones, y una sesión no es necesaria. A diferencia del TCP, UDP no provee control de tráfico ni envía acuse de recibo. Este tampoco retransmite paquetes perdidos ni da garantía de confiabilidad. El UDP es un protocolo sin conexión que es usado por el TFTP (Trivial File Transfer Protocol) y SNMP.

La Capa de Aplicación

Los protocolos de la Capa de Aplicación y sus respectivos RFCs sin listados a continuación:

HTTP-RFCs 1945 y 2616

El HTTP es utilizado para transportar documentos HTML (páginas Web) a través del Internet. El HTTP requiere de un programa cliente (un navegador) y un servidor, ambos ejecutando TCP/IP. El HTTP establece la sesión del servidor Web y transmite páginas HTML a un cliente con un navegador. El protocolo HTTP 1.0 establece una nueva conexión por cada página requerida, lo cual creaba mucho tráfico innecesario de Internet. El HTTP 1.1 establece las conexiones persistentes, lo cual permite multiple descargas con una sola conexión. Ambos el cliente y el servidor deben soportar el HTTP 1.1 para poder beneficiarse de estas ventajas que ofrece esta nueva versión.

FTP-RFC 959, STD 9

FTP es un sistema para transferir archivos entre computadores ejecutándose en red TCP/IP. El FTP ofrece una manera eficiente y rápida de transferir archivos ya que este no tiene que consumir los recursos de codifica y decodificar la data, como es enviar archivos de correo electrónico con archivos adjuntos. El FTP permite que archivos sean copiados al servidor, mientras que HTTP sólo permite que los clientes descarguen desde el servidor.

TFTP-RFC 1350, STD 33

El TFTP se usa para iniciar los sistemas sin discos (diskless). Funciona con BootP. El TFTP utiliza UDP, mientras que el FTP usa TCP. Debido a que el TFTP es simple y pequeños, este puede ser embedido en Memoria de Sólo Lectura (ROM),

lo cual es ideal para las estaciones de trabajo sin discos que buscan configuraciones a través de la red para iniciarse.

Telnet (remote terminal pro toco I)-RFCs 854 y 855, STD 8

Telnet es un protocolo de emulación de terminal desarrollado por ARPANET. El telnet le permite a los usuarios desde un sitio ingresar a otro remotamente y ejecutar programas desde el sistema remoto. El Telnet es usado normalmente por el terminal cliente ó un emulador de terminal vía software en una PC.

Network News Transfer Protocol (NNTP)-RFC 977

NNTP permite que sitios en el Internet intercambien artículos de noticias de Usenet, los cuales son organizados en tópicos como Programación en C++ o en Perl. Para usar los newsgroups, deberá tener acceso a un servidor NNTP que usted este autorizado para leer y postear noticias.

Gopher-RFC 1436

Gopher es un programa basado por menu usado para encontrar recursos en el Internet. Es muy similar a lo que hoy día es la Web. Los usuarios siguen de vínculos a vínculo de sitio a sitio en búsqueda de su información. Fué una de las primeras herramientas desarrolladas para establecer el acceso a los usuarios de todo el Internet y no sólo a un sitio. Los servidores Gopher han sido reemplazados por Servidores Web.

SMTP-RFC 821, STD 10

SMTP es el protocolo estándar de Internet para transferir mensajes de correo electrónico desde un computador a otro. Este específica como dos sistemas de correos deben interactuar. El SMTP es usado a menudo con el protocolo POP3 (Post Office Protocol 3), Cual es el estándar servidor de correo de Internet que utilizan el protocolo de mensajería del SMTP. El POP3 almacena e-mail entrantes hasta que los usuarios se autentican y lo descargan. POP3 está definido en el RFC 1939, STD 53.

SNMP-RFC1157, STD15

El SNMP es usado para administrar redes TCP/IP. Es un esquema estandarizado de administración que los fabricantes pueden soportar. Todos los dispositivos que cumplen los estándares SNMP pueden ser administrados por un administrador de SNMP. El SNMP también ofrece un bajo nivel de requerimientos de recursos, portabilidad, y son ampliamente aceptados.

Domain Name System (DNS)-RFCs 1034 y 1035, STD 13

El DNS es un mecanismo usado en el Internet para traducir los nombres de computadores host a direcciones válidas de Internet (IP). Es un método universal centralizado de resolución de nombre. Por ejemplo, cuando un usuario hace una petición de un FQDN (Fully Qualified Domain Name) www.abiertos.org, los servidores DNS traducen el nombre a la dirección IP 206.123.67.148.

BootP-RfCs 951, 1497 y 2132

BootP es una alternativa al RARP. El BootP permite que estaciones de trabajo sin discos no sólo determinen sus direcciones IP, pero parámetros adicionales, como lo son el gateway por defecto y las direcciones particulares de los servidores DNS. A diferencia del RARP que sólo provee una dirección IP.

Dynamic Host Configuration Protocol (DHCP)-RFC 2131

El protocolo DHCP está basado en el BootP. Como el BootP, este está diseñado para asignar direcciones de Internet y los parámetros adicionales, como son los gateways por defecto y los servidores DNS, a nodos en una red TCP/IP. A diferencia del BootP, las direcciones y los parámetros del DHCP pueden ser cambiadas con el tiempo (por esto es el nombre de dinámico). Los Servidores DHCP pueden temporalmente alquilar direcciones y parámetros por un tiempo específico a un cliente, entonces reasignar la información a otro cliente cuando el alquiler expire.

Multiplexado/Demultiplexing

Demultiplexing es el proceso que utiliza la computadora que recibe un paquete para poder procesarlo. Es una manera excelente de mostrar como es que el protocolo de Internet funciona dentro de la arquitectura de Internet. Aquí le mostramos una imagen para ilustrar este proceso.

99999999999999FOTO aquí

Protocolos Especializado de Interfase Serial

La gran mayoría de usuarios accedan el Internet desde su casa usando un modem. El Punto de Presencia (The Point of Presence, POP) es donde el usuario marca vía su modem para accesar el Internet. Este POP es normalmente su Proveedor de Servicios de Internet (Internet Service Provider, ISP). También es el punto donde el proveedor de larga distancia se conecta a la compañía de servicio local. Si la compañía local no existe, entonces el the POP es la línea conectada al usuario.

Conexiones vía modem son normalmente hechas sobre una línea telefónica estándar y utilizan el protocolo de Punto a Punto (Point-to-Point Protocol, PPP) ó el protocolo de Internet de Línea Serial (Serial Line Internet Protocol, SLIP) para conectarse a su ISP. En la siguientes secciones se describen estos protocolos.

Point-to-Point Protocol (PPP)-RFC 1661, STD 51

El PPP es un método de encapsulación para enviar paquetes a través de un enlace serial. fué creado en el año 1991 por la IETF y soporta ambos tipo de enlaces asincrónicos y sincrónicos. Debido a esto es que se puede ejecutar sobre líneas telefónicas estándares, enlaces full-duplex como son las líneas de Redes Digitales de Servicios Integrados (Integrated Services Digital Networks, ISDN), y las líneas de alta velocidades como las T1 y T3.

El PPP utiliza el protocolo de Control de Enlace (Link Control Protocol, LCP) para establecer, configurar, y poner a prueba una conexión durante el proceso de login (ingreso al sistema). Este protocolo permite que ambas computadoras puedan negociar y provee un alto nivel de confiabilidad. El PPP también hace disponible la protección de contraseñas utilizando los protocolos Autenticación de Contraseñas (Password Authentication Protocol, PAP) y el CHAP (Challenge Handshake Authentication Protocol).

El PPP tiene una familia de protocolos específica Network Uyet protocols llamados Protocolos de Control de Redes (Network Control Protocols, NCP). Los NCP existen para IP, AppleTalk, y el DECnet. Un ejemplo es, el NCP de IP que permite a un host negociar los encabezados de compresión headers.

Multilink Point-to-Point Protocol (MPPP)-RFC 1990

Si un usuario se conecta a su ISP usando una línea estándar ISDN, el PPP por lo normal utiliza un canal B de transmisión de 64-Kbps. Para obtener velocidades más alta de transmisión, dos ó más canales B pueden ser compartidos usando MPPP. Un ejemplo es, dos canales B de tipo ISDN de 64-Kbps pueden ser combinados para transmisiones a velocidades de 128 Kbps.

Serial Line Internet Protocol (SLIP)-RFC 1055, STD 47

SLIP es una forma simple de encapsulación para enviar paquetes IP sobre líneas seriales. SLIP puede ser usado en puertos seriales RS-232 y es usado comúnmente para conectar usuarios domésticos (no empresariales) al Internet desde líneas telefónicas estándares. SLIP soporta enlaces asincrónicos. Scripts ya automatizados son por lo general utilizados para agilizar el proceso de ingreso al sistema ó login. SLIP ha sido reemplazado por PPP ya que este es más rápido y confiable.

ENRUTAMIENTO DE TRAFICO

El proceso de identifica la ruta a una conexión a través del laberinto de las posibles rutas de conexiones en el Internet serán discutido en esta sección. Los paquetes deben ser enrutados de estación a estación hasta arribar a su destino. En esta sección discutiremos los protocolos que proveen el mecanismo para para la selección de rutas. Estos serán los tópicos a discutir:

- Protocolos Enrutables y No Enrutables/Routable and Nonroutahle Protocols
- TCP/IP
- IPX/SPX
- NetBEUI

- Redes de Multi-Protocolos/Multiprotocol Networks

Protocolos Enrutable y los No rutables

Algunos protocolos pueden pasar de LANs a WANs y más allá porque estas pueden direccionar y atravesar los enrutadores. Dentro de los protocolos enrutables están TCP/IP y IPX/SPX. Dentro de los protocolos no enrutable se incluyen el NetBEUI, el de Digital Equipment Corporation de nombre LAT (Local Area Transport), y el protocolo DLC (Data Link Control).

TCP/IP

El 1 de enero, 1983, las principales redes que conforman el Internet adoptaron el conjunto de herramientas (el Suite) TCP/IP como el protocolo oficial del Internet. Una de las razones del crecimiento explosivo del Internets y sus habilidades poderosas fué su adopción de esta suite, la cual originalmente fué desarrollada en la Universidad de Berkeley, en California.

Actualmente, el Internet soporta completamente la versión 4 del TCP/IP. La versión 6 del TCP/IP (mejor conocida como IPv6) ya se probó y ha adquirido soporte completo y esta siendo implementado en la actualidad. Discutiremos en detalle todo acerca del TCP/IP en todo este libro, algunos de los principios básicos lo cubriremos en esta sección.

Todo una Colección de Protocolos

El suite TCP/IP, es un conjunto de protocolos que incluye el Transmission Control Protocol (TCP), y el Protocolo de Internet (Internet Protocol, IP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), Address Resolution Protocol (ARP), y muchos más. Existen un core de más de 20 protocolos en el suite completo, cubriendo todos los aspectos del uso y la administración de las redes. Cada una de estos protocolos tiene una función en específico. En esta sección sólo discutiremos el TCP y el IP.

Transmission Control Protocol (TCP)

El TCP es el Protocolo de Control de Transmisión que garantiza una comunicación confiable y utiliza puertos para transportar paquetes. Además también fragmenta y reemsambla los mensajes, usando una secuencia de funciones para asegurarse que los paquetes sean reemsamblados en el orden correcto.

Internet Protocol (IP)

El IP es un protocolo sin conexión responsable de proveer direccionamiento de cada computadora y ejecución de enrutamiento. La Versión 4 del TCP/IP usa direcciones de 32-bit. El esquema de dirección cae en 5 clases, solo tres de cada uno esta disponible por el direccionamiento de redes estándares. El plan original fué asignar direcciones de clase A a grandes redes, Clase B a redes de tamaño, y Clase C para pequeñas redes. Las direcciones de Clase D son usadas para multicasting, y las direcciones de clase E son experimentales. (Más adelante se discutirá más sobre las clases de direcciones.)

Direcciones IP de 32-bit están divididas en dos partes: la porción de red y la porción de host. La mascara de subred ayuda a determinar cuales bits son de subred y de hosts.

Un estándar Abierto

El TCP/IP no esta atado a ningún vendedor y, por lo tanto, permite redes heterogéneas se comuniquen eficientemente. Las implementaciones están disponibles para un gran rango del sistema, al extenderse desde las redes de computadoras, Asistentes Digitales Personales (PDAs), laptops, y de computadores personales a mainframes. Este usa el modelo de arquitectura de Internet que divide este protocolo en cuatro capas. Cada capa es responsable de una tarea de comunicación y coinciden con las capas de OSI.

- Capa de Aplicación

• Capa de Transporte
• Capa de Internet
• Capa de Acceso a la Red

El TCP/IP en Acción

Una de las responsabilidades del IP es enrutar la información entre dos host entrelazados por Internet. Esta tiene el conocimientos de cual host enviarle los datos acortando la distancia de la manera más eficiente.

Foto de dos computadora conectadas mostrando las capas del tcp/ip

Si los dos puntos se encuentran físicamente en la misma red, entonces la comunicación es mucho más director; y no hay necesidad de la funcionalidad de enrutar.

TCP/IP Sobre GNU/Linux y UNiX

En sus inicio el desarrollo del TCP/IP fué muy cercano al de BSD UNiX, y la primera implementación apareció en la versión 4.2 BSD UNiX al principió de los 80s. La razón de esto es que ambos códigos fuente, el de UNiX y las especificaciones del protocolo TCP/IP, estaban disponible en toda la comunidad académica de Norteamérica. Los investigadores combinaron los protocolos TCP/IP con el sistema operativo UNiX para crear lo que se denominó un sistema operativo listo para el Internet.

Desde ahí al día de hoy la popularidad del TCP/IP ha crecido a proporciones gigantescas, y se ha convertido en el sistema de redes por defecto de UNiX. Combinandolo con la capacidad y el soporte que UNiX ofrece de Multiprocesadores Simétricos (Symmetric Multiprocessing, SMP) y su característica de multitarea (multitasking) hacen una combinación poderosa. De hecho, hoy en día todas las versiónes de UNiX vienen con TCP/IP. Los sistemas UNiX son en la actualidad la plataforma más usada como servidores de páginas Web.

UNiX es reconocido por su confiabilidad y es escogido las mayoría de las veces para los sistemas que ejecutan aplicaciones de misión crítica. En el pasado, entre las desventajas de UNiX se destacaban dos una era su alto costo y la otra era su falta de lo denominado userfriendly ó fácil uso. Pero recientemente, esto ha ido cambiando, clones de licenciamiento libre y/o abierto de UNiX, de los cuales GNU/Linux es sin lugar a duda es la versión más popular. Estas versiónes free/opensource y con calidad de nivel paralelo a los UNiX tradicional, ayudan a combatir la situación de costo. Se han ido elaborando más y más aplicaciones gráficas de toda indole, que ayudan en la situación del fácil uso y rápido aprendizaje. Hasta en el á de administración ya existen varios front-ends ó GUIs gráficos que hacen de la administración básica de sistemas GNU/Linux tareas mucho más fácil que tradicionalmente.

GNU/Linux con su kernel monolítico Linux y con la pila TCP/IP es uno de los sistemas operativos más rápidos disponibles en la actualidad. Además el kernel Linux también tiene excelente soporte de procesadores SMP para equipos con multiprocesadores. Por estas razones es que servidores GNU/Linux se ha ido ganando tanto espacio en el mercado de servidores Web, además de otros tipos de aplicaciones intensas de redes.

IPX/SPX

La compañía Novell, Inc., desarrolló el una vez dominante protocolo de LAN y WAN. Similar al TCP/IP, el IPX/SPX es un suite de protocolos y no un sólo protocolo. Sistemas operativos de Microsoft también ofrecen soporte de IPX/SPX, aunque la compañía lo ha renombrado NetWare Link (NWLink) por razones legales.

El IPX es un protocolo de la capa de Red sin conexión el cual es responsables del reenvío de los paquetes (forwarding packets). Los sistemas que se comunican entre si a través de una red IPX/SPX deben saber el número de red del host remoto Así como la dirección del sistema en la red. Finalmente ellos deben saber el número de socket del programa con el cual ellos se comunican. El IPX permite cada una de esta actividades.

9999999999999999999999999

SPX

El SPX es un protocolo de la capa de Transporte que usa los Servicios proveídos por el IPX. Como este reside en la capa de Transporte, este se asegura el transporte de la data seguro y confiable y además administra las sesiones.

Ventajas y Desventajas

El IPX/SPX no es un protocolo neutral de fabricante. Fué desarrollado por Novell, usado mayormente en redes de Novell NetWare. El TCP/IP ha desplazado totalmente el IPX/SPX como el protocolo estándar de las redes empresariales debido a su naturaleza abierta. Pero, el IPX/SPX está aún en uso y todo profesional de las redes e ingeniero de sistemas deben aún conocerlo, recuerde que siempre a presentado un mejor perfomance que el TCP/IP.

Aunque el IPX/SPX no e soportado por el Internet, existen miles de redes WAN ejecutándose sobre redes IPX/SPX privadas ó Redes Virtuales Privadas (Virtual Private Networks, VPN) para comunicarse sobre largas distancias.

Ya Novell ha adoptado el TCP/IP como su protocolo por defecto en las versiónes de Novell NetWare 5, aunque aún soporta el protocolo IPX/SPX.

NetBEUI

El protocolo NetBEUI es un acronismo de Network Basic Input/Output System (NetBIOS) Extended User Interfase. Primero fué desarrollada por IBM, más luego Microsoft la implementó como solución para sus redes peer-to-peer. Es un protocolo no enrutable, lo cual limita su uso en muchas redes.

NetBIOS

El NetBIOS fué originalmente diseñado para ser usado con NetBEUI (por esto lo del nombre NetBIOS Interfase de Usuario Extendido). Como el NetBEUI esta disminuyendo su popularidad, NetBIOS es usado principalmente como una interfase de de programación de aplicaciones. Este reside en la capa de Sesión (Layer 5) del modelo OSI. El NetBIOS puede operar sobre el NetBEUI Así como también como con protocolos enrutables como es el TCP/IP y IPX/SPX. Los equipos ejecutando Microsoft Windows usan nombres NetBIOS para identificarse entre si en la redes.

Redes Multiprotocolo

Las redes comúnmente usan protocolos enrutables como son el TCP/IP y el IPX/SPX, aunque esta combinación puede causar problemas con sobrecarga de sistemas, en sitios webs muy visitados. Tales combinaciones de sistemas proveén redundancias y pueden así acelerar su velocidad.

Hay veces que es beneficioso combinar protocolos enrutable y no enrutable , aunque se encuentre en una red enrutable. Un protocolo no enrutable como lo es NetBEUI puede resultar muy útil en situaciones LAN y WAN ya que este puede transportar tráfico a una computadora local sin la problemática asociadas del TCP/IP. Si un usuario envía un mensaje a un empleado en la misma LAN, entonces el NetBEUI controlará

toda esta transacción. pero, si alguien envía un mensaje a alguien en otra LAN (actividad que obviamente involucra un enrutador), El sistema automáticamente usará un protocolo como lo es el TCP/IP.

RESUMEN

En este capítulo discutimos los siguientes tópicos de redes:

- El concepto de redes y su rol en la intercomunicación de data en las empresas hoy día
- Como se relacionan el Internetwork (trabajo a través del Internet) al concepto de redes corporativas empresariales
- Servidores, estaciones de trabajos, y hosts
- características Básicas de las topologías de Redes
- Sistemas Operativos de Redes (NOS)
- Puntos básicos de las Redes de Area Local (LAN) y las Redes de Area Amplia (WAN)

PREGUNTAS POST-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿De cuantas capas consiste el modelo OSI? ¿Cual es la capa que más se relaciona con hardware?

2. ¿Que son los RFCs?

3. Liste tres protocolos comunes del Internet y los puertos que generalmente usan.

FUNDAMENTOS DE IP

OBJECTIVOS

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Usar el TCP/IP para efectúar funciones básicas.
- Definir la importancia de los archivos de inicio del sistema y los pasos necesarios para accesarlos.
- Entender el rol de los scripts y los archivos de configuración en /etc/sysconfig/network-scripts.
- Configure y use el Protocolo PPP (Point-to-Point Protocol).
- Describe el sistema de direcciones de IP así como el nuevo sistema que ha sido desarrollado para poder enfrentar el crecimiento exponencial del Internet.
- Describa la herramienta de red WvDial y su rol en la configuración de una red.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuál es la función del daemon inetd?

2. ¿Cuáles son los tres tipos de topologías lógicas de redes?

3. ¿Cuáles dos protocolos pueden ser usados para conectarse al Internet vía modem? ¿Cuál es mejor, y por qué?

4. ¿Cuál es el rango de las tres principales clases de direcciones?

5. ¿Por qué existe la necesidad del IPv6?

INTRODUCCIÓN

El Protocolo de Control de Transmisión (Transmission Control Protocol/Internet Protocol, TCP/IP) es el protocolo principal de redes usado por todos los sistemas GNU/Linux. Es utilizado por todas las computadoras que desean comunicarse en el Internet con otras computadoras. También puede ser usada para intercomunicar computadoras en redes privadas e intranets.

Para que los host se puedan comunicar con otros hosts remotamente sobre el Internet, deberá saber la dirección de Internet. Cada host, ó nodo, tiene su propia dirección Internet de 32-bit, ó dirección IP, que la identifica como equipo único y diferente de cualquier host en el Internet.

Este capítulo se inicia describiendo las capas con que cuenta el TCP/IP: Las capas Física y la de Enlace de Data del modelo OSI. Identificaremos los componentes necesarios que permiten que LANs y WANs funcionen. Luego describiremos las tecnologías de redes más comunes como son el Ethernet, Token Ring, PPP, y Serial Line Internet Protocol (SLIP). También cubriremos la versión actual del protocolo de Internet (IP) para aplicar direcciones de Internet y las mayoría de redes TCP/IP de hoy, el IPv4, y el del futuro el IPv6. La estructura de las direcciones de Internet, el formato binario contra el formato decimal, clases de direcciones, reglas de direccionar, direcciones reservadas, subredes (subnetworks), mascaras de subredes, y rango de direcciones.

COMPONENTES DE REDES COMÚN

Redes de computadoras por lo normal requieren un conjunto de equipo para poder funcionar apropiadamente. Típicamente las LANs y WANs incluyen Tarjetas de Interfase de Red (Network Interfase Cards, NICs), repetidores, hubs, bridges, enrutadores (routers), brouters, y switches.

Cubriremos los siguientes componentes en esta sección:
- **Network Interfase Cards (NICs)**
- **Repeaters**
- **Hubs**
- **Bridges**
- **Routers**
- **Brouters**
- **Switches**
- **Gateways**
- **Channel Service Unit/Data Service Unit (CSU/DSU)**
- **Modems**
- **Patch Panels**
- **Internet in a Box**

Network Interfase Cards (NICs)

Cada nodo en la red debe tener una tarjeta llamadas NIC, a menudo referenciadas como tarjeta de red. El NIC es el interfase entre el computador y la red. Hoy día la mayoría de tarjetas madres ó motherboards, la traen integrada, pero es más común verla como tarjetas de expansión PCI. Estas se comunican con otras computadoras a través del NIC y su manejador. Redes se conectan a través de un cable conectado a la NIC y de ahí a la red. Los NICs varían en su tipo de tecnología como son redes Ethernet y Token Ring, las cuales estudiaremos en el próximo capítulo. Las NICs operan en la capa de Enlace de Data del Modelo de OSI.

La mayoría de los NICs contienen un transductor, un dispositivo de red que transmite y recibe señales análoga ó digital. La terminología es de transmitor y recibidor. En las LANs, los transductores colocan data

en el cable de la red y detecta y recibe la data que se traslada a través del cable. Algunos tipos de redes requieren un transmitor externo.

Repeaters

Un repetidor es un dispositivo de bajo nivel que amplifica la señal electrónica que viaja en el segmento de un cable. Nos asegura que la señal electrónica no degrade. así que un repetidor puede conectar dos computadoras que se encuentra a una distancia mayor que la definida por los Estándares, así como Ethernet.

Los repetidores operan en la capa Física (Layer 1) del modelo OSI. Ellos transmiten código binario, los cuales consisten en unos y ceros. Un segmento de cable que agota su longitud máxima empieza a degradar su señal y eventualmente se colapsa. Un repetidor puede reforzar la señal a través de su retransmisión.

Hubs

Un hub conecta las computadoras en una configuración de red tipo estrella (star) para que ellas puedan intercambiar información. Tienen varios puertos disponibles, cada uno conectado a un nodo único. a través de esta interconexión de los nodos, un hub sirve como el punto de concentración de una red. La mayoría de los hubs son conocidos como hubs activos porque ellos regeneran señales electrónicas al igual que los repetidores.

Los hubs operan en la capa Física (Layer 1) del modelo OSI. Ellos se pueden conectar a otro hubs, y conectados en cadena para poder tener más puertos disponibles para acomodar redes más grandes. Usted puede conectar hubs a los switches ó a los routers para incrementar el número de nodos.

Bridges

Los bridges son dispositivos que filtran las tramas (frames) para determinar si un frame en específico pertenecen a un segmento local ó a otro segmento de LAN. Como los bridges operan en la Capa 2 del modelo OSI, ellos usan direcciones de hardware para determinar cual segmento recibirá la trama.

Los bridges pueden reducir el tráfico de la red dividiendo una red en dos segmentos. Ellos también pueden conectar segmentos de redes con el mismo ó diferente protocolos de enlace de data, habilitandolo para comunicarse. Por ejemplo, un bridge puede conectar una red de Ethernet con una de Token Ring ó conectar dos redes Token Ring.

Los bridges reconocen hardware, ó direcciones de Control de Acceso al Medio (Media Access Control, MAC), entre las redes. Supongamos que un computador envía información a otra en una red tipo Ethernet. El bridge determina si la computadora de destino reside en el mismo segmento de red a través de la verificación de la dirección MAC. Si el computador de destino reside fuera del segmento, el bridge pasa el mensaje a otro segmento. a través de este proceso de filtrado, el tráfico de la red es reducido y se libera más ancho de banda y se hace disponible.

Los bridges son independiente de todos los protocolos de las capas superiores. Esta independencia permite a los bridges poder reenviar tramas que provienen desde muchas diferentes capas de protocolos.

Enrutadores/Routers

Los routers son conceptualmente similar a los bridges excepto que estos obedecen la capa 3 del modelo (Layer 3) OSI. En ves de usar la dirección MAC, los routers usan protocolos de redes, como es el IP y el IPX. Ellos reenvían, o enrutan, data desde una red a otra en vez de un segmento a otro.

Los routers direccionan los paquetes de data entre las redes. Ellos identifican la dirección de red de la

computadora de destino, y entonces determinan la ruta más eficiente para transmitirle la data.

Suponga que una máquina en la red 2 envía data a otra máquina en la red 2. En este caso, el router no transmitirá la data a la red 1. Este proceso de filtrado conserva en ancho de banda de la red. Se puede medir la capacidad de un router a través de su rango de paquetes por segundo (packets per second,PPS).

Los routers dependen del protocolo—ellos dependen del sistema de direccionamiento definido por el protocolo usado (IPX, IP, entre otros). Diferentes tipos de routers trabajan con diferentes protocolos. Por ejemplo, los IP routers operan con la estructura direccional del inherente IP de 32-bits. Para usar el protocolo de direccionamiento IPX, es necesario un router que soporte tanto IP como IPX. El enrutamiento se explicara com más detalle en este capítulo.

Brouters

Los brouters operan en las capas 2 y 3 del modelo OSI. Ellos incorporan la funcionalidad de los bridges y routers a la vez. Los brouters son protocolos independientes y pueden enrutar el tráfico de varias redes. Por ejemplo, ellos pueden enlazar paquetes DECnet y enrutar un paquete TCP/IP.

Switches

En una red, un interruptor (switch) direcciona el flujo de información de un nodo a otro. Los Switches operan más rápido que los dispositivos de red tradicionales, tales como los hubs, bridges, y routers, y están progresivamente reemplazando estos dispositivos. Un switch puede asignar a cada par de emisor/receptor el ancho de banda total de las líneas en lugar de compartir el ancho de banda con todos los demás nodos de la red.

Los switches ofrecen los siguientes beneficios para las redes Ethernet y Token Ring:

- **Instalación sencilla**
  Para la mayoría de los bridges y hubs, al instalar un switch se requiere que usted desconecte desde los dispositivos existentes y luego reconecta todos los dispositivos en los puertos del switch.
- **Alta Velocidades**
  Los witches tienen backplanes de alta velocidad (que son las conexiones dentro del switch mismo) que permiten banda ancha completa entre cualquier dos usuarios o cualquier otro dos segmentos. Esta característica elimina el switch como un potencial cuello de botella en la red.
- **Más Ancho de Banda en los Servidores**
  Los servidores se pueden conectar directamente a los switches. Esta capacidad permite que los usuarios de la red usen todo el ancho de banda al accesar los recursos del servidor.

Los switches pueden operar en varias capas del modelo OSI. Un switch de Capa 1, llamado un hub de switching, sirve para reemplazar los tradicionales y mucho más lento hub. Un switch de Capa 2, también llamado un LAN switch, reenvía tráfico basado en la dirección MAC. Un switch de Capa 3 reenvía tráfico basado en la información de la capa 3y se llama un routing switch y soporta protocolos de red, como son IP y el IPX. Los switches de Capa 4 toman desiciones de reenvío basadas en la información de capa 4, como puede ser el puerto específico TCP/UDP que una aplicación usa, así como la información de las capas 2 y la 3.

Las opciones para manejar un incremento del tráfico de red de una LAN

A continuación tres opciones que pueden manejar el incremento de tráfico en la LAN:

- **Usar un bridge.**
  Este método tradicional reduce el número de usuarios en una red a través de separarla en dos segmento de red.

- **Usar un LAN switch (un switch de Capa 2)**
  Los LAN switches están disponibles para Ethernet, Fast Ethernet, Token Ring, y FDDI (aprenderemos más de estos Estándares de red más adelante en el próximo capítulo).

• Incrementar el ancho de banda de la red

Una forma de incrementar ancho de banda es mudarse hacia Estándares de más alta velocidad como es el estándar FDDI. Para lograr estos cambios hay que cambiar adaptadores, re-alambrar todas las conexiones, y es muy posible tener que cambiar software del sistema. Actualizar a FDDI es costoso y en algunas situaciones hasta caro. Actualizar de Ethernet a Fast Ethernet es mucho más económico. Aprenderemos de FDDI y Ethernet en este capítulo.

Gateways

Los Gateways, que también son conocidos como convertidores de protocolos, pueden operar en cualquier capa del modelo OSI. El trabajo de un gateway es mucho más complejo que ese de un router o switch. Normalmente, un gateway deberá convertir desde una pila de protocolo a otra. Por ejemplo, un gateway puede interconectar un nodo de red AppleTalk a un nodo DECnet o a un nodo de red TCP/IP o a un nodo de red IPX/SPX. Discutiremos otro tipo de gateway, el gateway por defecto, más adelante.

Channel Service Unit/Data Service Unit (CSU/DSU)

Un CSU/DSU (Unidad de Canal de Servicio/Unidad de Servicio de Data) es donde terminan las conexiones físicas. Este dispositivo es requerido cuando usamos circuitos dedicados, como son las líneas T1. El flujo digital de data es traducido por el CSU/DSU a señales bipolares, las cuales son adecuadas para la transmisión por línea.

Por ejemplo, una línea dedicada Tl entra a un edificio a través de un conector RJ-45 (Registered Jack-45), el cual es parecido a un conector de teléfono pero más grande (los jacks de teléfonos usan conectores RJ-11). El CSU/DSU entonces transmite la señal a la red.

El CSU/DSU también lleva acabo cierto reporte de errores y funciones de loopback. Los CSU/DSUs opera en la Capa Física (Capa 1) del modelo OSI.

Modems

Un modem es un dispositivo que permite que un computador se comunique con otras computadoras sobre una línea telefónica a través de la traducción de data digital a señales audio/análogas (en la computadora que envía) y de nuevo a data digital en la computadora que la recibe. Este tipo de modem es denominado tradicionalmente como modem análogo o simplemente modem.

El término modem es usado en una manera muy general y no siempre se refiere a modems que traducen análogo-a-digital y vice versa, por ejemplo otros son modems del tipo cable, DSL y los ISDN que son usados en todas las redes digital; la traducción a análoga no es requerida ni necesaria. La terminología modem a sido usada para describir cualquier dispositivo que adapta una computadora a una línea telefónica o a redes de Telecable, aún sea digital o análoga.

Un modem puede ser compartido en una red con todos los usuarios con acceso a ella. La computadora que se encuentra conectada físicamente al modem debe estar configurada apropiadamente. Al configurar un modem para acceso acceso por una WAN (o cualquier tipo de actividad o propósito de red), deberá definir varios ítemes para asegurar un funcionamiento exitosos. Estos ítemes incluyen:

- IRQ del Puerto Serial
- Dirección de E/S (I/O Address)
- Velocidad máxima del Puerto

IRQ del Puerto Serial

Una línea de Interrupt Request (IRQ) es usada por componentes del tipo modems, tarjetas de redes, y los teclados para requerir atención del procesador del sistema. Los modems seriales pueden usar una de las líneas de IRQ 3 o 4, ambas son usadas para puertos seriales.

I/O Address

Las direcciones de Entra/Salida (input/output, I/O) transfieren información entre el CPU y un dispositivo en específico. La configuración de los puertos base de I/O de un modem son 3FO al 3FF para el COM1 y la 2FO a la *2FF* para el COM2.

Velocidad máxima del Puerto

Los modems están disponibles con diferente capacidades para enviar y recibir datos. Al configurar su modem, usted debe establecer la velocidad máxima del modem para asegurar que funcione apropiadamente. Esta configuración provee la conexión más confiable para el modem y de optima velocidad de transmisión para su sesión.

Patch Panels

Los Paneles de Patch agrupan los sockets (normalmente consisten en pin de conexiones y puertos) montados en un rack. Es un punto central donde los diferentes cables desde los diferentes puntos de la red (digamos los departamentos) pueden ser interconectado (así formando una LAN). Puede ser usada para conectar una red completa al Internet o a otra WAN.

Los patch panels son generalmente colocados en un punto central, en sitios como en un armario o en cuarto de máquinas de la compañía. Ellos pueden tener varios puertos y los pines de conexión dependiendo del tamaño de la compañía.

En un lado el patch panel consiste de filas de pines de conexión. Los cables que se conectan a los pines de conexión se introducen con una herramienta de empuje (punch tool) para hacer la conexión. Estas conexiones por lo general se originan desde jacks en las paredes de la oficina. Por ejemplo, el departamento nómina con 35 estaciones de trabajo puede conectar cada estación de trabajo a un hub. El hub entonces es conectado a un jack en la pared, el cual es entonces conectado al patch panel.

El otro lado del patch panel contiene una fila de puertos de conectores hembra, los cuales son usados para conectarse a otros dispositivos de red, somo son los routers y los switches. Por ejemplo, el patch panel puede ser conectado a un router, el cual entonces es conectado al Internet.

Los Patch cords son usado en los puertos para conectar crusado computadoras en red que están conectadas al patch panel. Los patch cords conectan dispositivos de red a un jack de pared (wall jack).

Listo para el Internet

GNU/Linux ofrece distribuciones que ofrecen la oportunidad, a pequeñas empresa y usuarios, de soluciones toda incluido de Internet. Distribuciones especializadas que ofrecen todo el software necesario para el acceso a Internet. También existen productos llamados soluciones TurnKey que incluyen además del software también el hardware. Por ejemplo estas soluciones pueden que incluyan servidores de correo, páginas web entre otras cosas. Este tipo de soluciones son de las más demandadas en comunidades en paises donde se empieza a desarrollar el uso de GNU/Linux y aún existen pocos expertos.

MEDIO DE TRANSMISION

Para transmitir data, debe existir un medio físico, generalmente un cable o por el método común de hoy día wireless. En esta sección explicamos los tipos de cables más comunes:

• Cable Twisted-Pair (Trenzados)

• Cable Coaxial
• Cable de Fibra-Óptica
• Medio Inalámbrico/Wireless Media

Cable Twisted-Pair

El tipo de cable Twisted-Pair es quizás el más usado sistema de cable en las redes de Ethernet. Dos alambres de cobre trenzados entre si para formar un par de cables entre-tejidos. Dependiendo de la categoría, varios hilos de alambre aislados pueden residir en un sólo cable. Los diferentes tipos de categorías de cable serán discutidas más adelante en este capítulo.

Un segmento de cable trenzado no puede sobrepasar los 100 metros. Los cables Trenzados son usados en muchos de los Estándares de redes. Por ejemplo, Redes Ethernet 10BaseT usan cable Trenzados; el nombre 10BaseT significa una red transmitiendo a 10 Mbps usando transmisión en banda base y cable par-trenzado. Los cable trenzado están disponibles en dos formas básicas:

Trenzados Blindados/Shielded Twisted Pair (STP)
Este es un trenzado de alambre de cobre protegido de interferencia electromagnética externa por un blindado de metal que cubre todo el alambre; es más dificil de instalar y mantener que el UTP.

Trenzados sin Blindaje/Unshielded Twisted Pair (UTP)
El tipo de cableado UTP es el más común de los cables trenzados; es más económico que el STP pero menos seguro y subsetible a las interferencias electromagneticas. En este libro sólo nos concentramos en el UTP.

Alambre STP y UTP están disponible en dos variedades:

• **Stranded**
Este es el tipo más común; es flexible y fácil de manejar en las esquinas y los objetos.

• **Solid/Sólido**
Este tipo puede usarse en distancias más grande sin desarrollar perdidas como el del tipo stranded, pero es menos flexible; se romperá si se dobla multiple veces.

Hay cinco Estándares de cables trenzados especificados por la Telecommunications Industry Association/Electronic Industries Alliance (TIA/EIA) 568 Commercial Building Wiring standard. Dos niveles adicionales, las Categorías 6 y la 7, también son comerciales pero no estandarizadas aún (la 6 ya si lo es) pero son usadas comercialmete. En la siguiente tabla se definen estas categorías.

Categoría	Descripción
1	Usado para voz, no para data (UTP only)
2	Contiene cuatro trenzados y una transmisión de data hasta 4 Mbps; usado para redes Token Ring (Sólo UTP)
3	Contiene cuatro trenzados y una transmisión de data hasta 10 Mbps; usado para Ethernet
4	Contiene cuatro trenzados y una transmisión de data hasta 16 Mbps; usado para redes Token Ring
5	Contiene cuatro trenzados y una transmisión de data hasta100 Mbps; usado para Ethernet y Fast Ethernet y permite que Ethernet sea fácilmente actualizada a Fast Ethernet
6	Contiene cuatro trenzados y una transmisión de data hasta 155 Mhps; usado para Fast Ethernet
7	Contiene cuatro trenzados y una transmisión de data hasta 1,000 Mbps; usado para Ethernet de Gigabit

Conectores RJ-45

Los conectores RJ-45 son comúnmente usados ciertos tipos de redes Ethernet y Token Ring, las cuales discutiremos más adelante en este mismo capítulo. Los conectores pueden sostener hasta ocho alambres y es usada con alambres trenzados. Para colocar un conector RJ-45 a un cable, debe hacerse usando una herramienta especial llamada crimper tool. Para colocar el conector, primero debe poner el cable con los alambres en posición correcta (la posición del alambre depende en cual de los estándar esta usando, lo discutiremos más adelante en este mismo capítulo). Luego coloque el conector en la herramienta crimper incluyendo el cable y simplemente oprima la herramienta firmemente.

El crimper empuja dos grapas desde el conector RJ-45 dentro del cable. Una grapa se inserta dentro del exterior del cable (el jacket) y así establece una firme conexión, entre conector y el cable. La otra grapa empuja ocho pines trapazando el cobertor del cable y lo introduce dentro del conector cada respectivo alambre cayendo en una canaleta.

Un conector RJ-45 es un poco más grande que un conector estándar de teléfono RJ-11, el cual solo puede recibir cuatro alambres. (Existe un RJ-11 de seis alambres también.) Un conector de 50 pines RJ-21, es un conector normal RJ-45 de un lado que se expande a doce (12) conectores RJ-45 del otro lado.

Coaxial Cable

El Cable Coaxial, mejor conocido como coax, es un cable de alta capacidad usado para redes de video y comunicación. Este proveé anchos de banda más altos que eso proveídos por cables par-trenzados. Los cables Coaxiales contienen en el centro un cable de señal, el cual puede ser hebras tejidas (stranded) o solid, envuelto el una malla de protección metálica que a la vez le sirve de tierra. Estas protecciones son o del tipo malla tejidas o solidas y son envuelta por una camisa plástica. Si el cable viaja por un plenum, este entonces es cubierto por un material a prueba de fuego, como es el Teflon.

Existen varios tipos de de cables coaxiales para diferentes propósitos. Por ejemplo, los cables coaxiales fueron diseñados para baseband, broadband, y redes de televisión.

Cable Coaxial Grueso/Thick Coaxial Cable (Thicknet)

El cable Thick Coaxial, o Thicknet, es considerado el estándar de Ethernet. Es muy a menudo referido como el cable amarillo, aunque el cable esta disponible en una gamma de colores. El nombre Thicknet se refiere a a uno de los varios métodos de conectar computadoras a un ambiente de red. Es usado para redes Ethernet 10Base 5 (Redes Ethernet funcionando transmisiones en una banda de 10 Mbps y cable grueso coaxial de 1/2 pulgadas de diámetro).

El cable Thicknet trabaja bien en ambientes donde interferencias de magnetismo pueden afectar el cable; su uso es muy común en hospitales donde se usan equipos de Computadoras para asistir diagnóstico como son los CADScan y los equipos de resonancias magnéticas o scanners para producir imágenes (MRI y CT). El cable no es muy flexible, por esto dobla fácil en esquinas. Cables Thicknet pueden ser tendidos en segmentos de un máximo de 500 metros.

Cable Coaxial Fino/Thin Coaxial Cable (Thinnet)

El cable Thin coaxial, o Thinnet, es un estándar Ethernet para redes pequeñas. El Thinnet es altamente flexible y trabaja a perfección en á reducidas, como son oficinas divididas por cubículos. Un segmento Thinnet puede ser tendido hasta un máximo de 185 metros, pero con dispositivos de de red (como es un repetidor) esta distancia puede ser extendida. Thinnet es usado para redes Ethernet 10Base2 (Redes Ethernet ejecutándose transmisiones sobre una banda de 10 Mbps y un cable thin coaxial de 1/4 pulgada en diámetro).

Conector BNC

El conector BNC (British Naval Connector o Bayonet Neill-Concelman, BNC) es comúnmente usado para conectar cables coaxiales a NICs, hubs, y otros dispositivos de red. El BNC es engrapado al cable usando una técnica de nombre bayonet mount. La técnica bayonet mount conecta dos alambres (señal y tierra) en el cable coaxial al conector. El conector es luego insertado dentro de otro conector y luego enroscado, lo cual causa que el mecanismo bayonet pinche varios pines en la ranura del BNC.

Cable de Fibras-Ópticas/Fiber-Optic Cable

Cable de fibra-Ópticas pueden acomodar transmisión de data a velocidades muchas más alta que las de cables coaxiales o par-trenzados. Líneas de fibra-Ópticas pueden transmitir data a velocidades de en los rangos de gigabits por segundos. Ya que ellas envían data como pulsos de luz sobre hebras de vidrio, la transmisión puede viajar millas sin degradación de la señal. Como es un rayo de luz no se transmite ninguna señal eléctrica sobre la fibra-Ópticas, así que las líneas son libre de interferencias electromagnéticas y son extremadamente difíciles de interferir.

Cables de fibras-Ópticas consisten en dos hilos de vidrio pequeños; uno que recibe y el otro que envía. Estas hebras de hilo son llamadas el núcleo (core), y aveces son echas de plásticos. Cada core es rodeado por vidrio cladding. Cada core y elemento cladding está envuelto con una fibra Kevlar de plástico reforzado.

Transmisores láser envían los pulsos de luz modulada, y recibidores ópticos los reciben. Existen dos principales tipos de cables de fibra Ópticas:

• **Single mode**

Este usa una longitud de onda específica de luz (wavelength). El diámetro del core de este cable es de 8 a 10 micrónes. Fibras de Mono Modo es muy comúnmente usado en troncales telefónicas en el interior de ciudades y aplicaciones de video.

• **Multimode**

Este usa un número mayor de frecuencias (o modos). El core del cable es más grande que el del mono modo. Fibras multimodo es el tipo normalmente especificado para las LANs y WANs.

Cables de fibra Ópticas es costoso y requiere un profesional para su instalación y las conexión de dispositivos de redes. Con el crecimiento de del uso de fibra Ópticas, esta progresivamente requerirá menos niveles de experto ya que los avances tecnológicos están simplificando el proceso de instalación y conexión.

Medio Wireless

Redes de comunicación inalámbricas se han convertido más y más popular cada vez. Redes inalámbricas son comúnmente implementadas en ambientes híbridos. Un ambiente híbrido es aquel en el cual componentes inalámbricos se comunican con redes que utilizan cables. Por ejemplo, un computador tipo laptop puede usar sus capacidades wireless para comunicarse con la red corporativa LAN que utiliza cables para sus conexiones.

La única diferencia entre una LAN inalámbrica y una LAN cableada es sólo el medio, y su NIC y el transceptor para cada computador wireless. El transceiver es a menudo llamado un punto de acceso (access point) ya que este recibe y envía señales desde y hacia la red.

El uso de un medio de transmisión wireless para conectar una LAN tiene un uso muy selectivo en las arquitecturas de redes disponibles a la fecha. La necesidad de una LAN wireless y la existencia de una amplia gama de diferente tipos disponibles. Conectar edificios a larga distancia, lo que permite a los usuarios conexiones rápida, portable o temporal a una LAN o permitir acceso a una LAN móvil, encierra algunas de las razones para la capacidad de wireless en una LAN. Al considerar el uso de conexiones wireless en una LAN, existen tres métodos de transmisión que son Típicamente usados: transmisión de micro onda (microwave), transmisión infrarrojo (infrared), y la transmisión ondas de luz (lightwave).

Transmisión Microwave

Desarrollada por la industria telefónica para transferir llamadas de larga distancia, la tecnología microwave ha sido incorporada las operaciones de redes LAN. El espectro de frecuencias usado por las compañías telefónicas esta regido por el FCC (Federal Communications Commission) en los EEUU. Pero existe un conjunto de frecuencias por separado, como es el Industrial/Scie999999999 mi fie/Medical band, que no requiere licenciamiento del gobierno. Además, las frecuencias en el rango de 902- a- 928-MHz son frecuencias no restringidos. Por esta razón, LANs wireless locales que usan un medio de transmisión wireless pueden considerar este espectro de frecuencia para las redes de poca extensión.

La habilidad de conectar dos usuarios sobre una distancia amplia es posible en un sistema de transmisión de microwave. Típicamente sistemas de transmisiones microwave le permiten a los usuarios conectarse desde distancias hasta de 12 millas. Existen varias ventajas de la capacidad de relativamente larga distancias de conexión inalámbrica. Un uso común de este sistema de transmisión es para conectar edificios separados por distancia a la localidad Física de la LAN. Por lo general instalar conexiones físicas entre las dos localidades tienen un enorme costo y se puede tornar impractico. Si las dos localidades o los sistemas de transmisión están en una misma línea de visibilidad, la posibilidad de proveerle interconexión vía wireless LAN es posible.

Existen varias desventajas que tornan la transmisión de microwave impractica o imposible. Primer, para una conexión de larga distancia, las bandas de Industrial/Cientifica/Medical no pueden ser usadas como espectro de frecuencia de transmisión, y restricciones sobre la salida de potencia no permitirá las señales radiales viajar lo suficientemente lejos para completar una conexión a una distancia larga. Una transmisión típica de microwave es llevada acabo con frecuencias que operan alrededor de 10 GHz. Con estas frecuencias, proveído de que suficiente energía es transmitida con el equipo correcto, la transmisión de data a una LAN es posible. Eso si, este espectro de frecuencia requiere licenciamiento por las autoridades locales, en el caso de los EEUU, la FCC. Dependiendo del tiempo que la conexión necesita, obtener el licenciamiento para utilizar el espectro de frecuencia específico puede ser que se torne no práctico o mejor dicho rentable, para una conexión LAN de larga distancia puede tornarse como una opción no viable y que sea mejor conectarse a través del Internet y convertice en una extranet.

Transmisión Infrarroja/Infrared

Transmisiones Infrarrojas tienen utilidad en conexiones LAN modernas. El uso de esta tecnología es para comunicaciones a corta distancia. El uso primario de este espectro esta en dispositivos de control remoto para dispositivos multimedia como son televisores y VCR.

Existen muchos beneficios con el uso de tecnología de transmisión Infrarroja. El infrarrojo es relativamente direccional, económico, fácil de construir y mantener. La instalación de una red Infrarroja no requiere los cables utilizados en una LAN típica. Además, la versatilidad de los usuarios poderse conectar dinámicamente en una LAN requiere sólo confirmación vía software, previsto que tengan todo el equipo necesario para tener infrarrojo propiamente instalado en el LAN. Pero claro, existen desventajas primordiales en la tecnología de transmisión Infrarroja. Debido a la frecuencia de transmisión, las señales Infrarrojas no pueden traspasar objetos sólidos. Causado por esta limitación, las LANs Infrarrojas esta restringida a conexiones cortas y que no existan obstrucciones entre el transmisor y el recibidor. Otro limitante adicional es que transmisiones Infrarrojas deben estar libre de la luz del sol ya que este transmite mucha energía Infrarroja. Estar expuesta a la luz del sol para este tipo de LAN la tornaría sin uso.

El uso primario de conexiones Infrarrojas en redes LAN es conectar usuarios mó y temporales a la LAN. Eventos y Conferencias en las cuales computadoras portátiles serán usadas por corto tiempo pueden usar este tipo de transmisión para conectarse sin físicamente tener que conectarse a la LAN a través del uso de un trans-

misor localizado en el centro.

Transmisión de lightwave (Onda de Luz)

Muy a menudo es necesario proveer una localidad remota con la capacidad de accesar una LAN en la sede principal. La disponibilidad de poder proveer conexiones físicas con el uso de cables no siempre es posible, dependiendo de donde se encuentra el sitio remoto. Este problemas puede ser resuelto a través del uso de un sistema de transmisión de micro-ondas o uno de lightwave. Las limitaciones de sistemas de transmisiones microwave ya la explicamos; pero, un sistema de transmisión de lightwave tiene características similares que deben ser entendidas para así poder elegir entre el sistema apropiado de transmisiones wireless.

Un sistema de transmisión lightwave consisten en un láser y un foto detector, normalmente colocado en sima de edificios. Para colocar este tipo de conexiones, el perímetro de alcance de este sistema de transmisión es de varias millas. Aunque los sistemas de transmisión de ondas de luz funcionan bien en LANs que cubren corta distancia, existen algunas desventajas claves. Los rayos de luz tienen problemas al transmitir a través de precipitaciones atmosféricas, similar a la falta de visibilidad en los automóviles cuando se maneja a través de neblinas espesas. Condiciones climáticas adversas tornaran las conexiones a través de transmisiones vía ondas de luz inservible. Adicionalmente radiaciones solares reflejan desde edificaciones traspasan el rayo láser a aproximadamente la misma frecuencia e interfieren con las conexiones láser. Al ocurrir esto, el láser es refractado por el calor y falla los foto detectores.

Satélite

El uso de Satélite a lo que respecta a redes LAN de datos, aún se encuentra en pleno desarrollo. La disponibilidad de usar satélites para conectar a usuarios remotos a la red principal es el uso principal de este tipo de conexión. El uso de satélites para proveer conexiones LAN aún se tornan un poco difícil debido a unas cuantas desventajas.

Las instalaciones y los equipos que involucra sistemas de conexiones vía Satélite son costosos. La instalación del sistema requiere equipos especializados a un muy alto costo. Otra desventaja es la distancia que los datos tienen que viajar. Existe un retraso que no se puede obviar de hasta medio segundo al usar conexiones vía Satélite.

Causado por los costos aún extremos asociados al uso de conexiones satelitáles para las operaciones de redes LAN, la disponibilidad de este tipo de conexiones es relativamente bajo. Su uso primordial es la de conectar a usuarios individuales remotos, a las redes locales LAN de las compañías. Esto va cambiando rápidamente y con el crecimiento de las redes satelitáles puede ser que cambie en un tiempo prudente.

TIPOS DE TRANSMISIÓN

Una vez establecida una red, los datos deben ser transmitidos por el medio elegido, en la mayoría de los casos es un cable. En esta sección discutimos algunos conceptos de transmisión de datos, que incluyen los siguientes:

- Transmisión Sincrónica
- Transmisión Asincroníca
- Flujo de Transmisión de Data
- Transmisiones Baseban y Broadband (Banda base y Banda Ancha)
- Topologías lógicas y Física

Transmisión Sincrónica

En transmisiones sincronizadas, el dispositivo de acceso y la red comparten el reloj y la velocidad de transmisión. A este tipo de transmisión se le llama sincronizada. Los datos son intercambiados a través de flujo de caracteres llamados tramas de mensajes de data (message-framed data). Una secuencia de inicio-y-pare (start-and-stop) es asociada con cada transmisión. Los dispositivos de acceso y de red necesariamente deben estar sincronizados para poder recibir el mensaje por completo en el orden que fué enviado o transmitido. Líneas T1 usan transmisiones sincronizadas.

Transmisión Asincrónicas

Transmisiones asincronizadas son caracterizadas por la ausencia de un reloj en el medio de transmisión. El dispositivo de acceso no está sincronizado con el dispositivo de red. Pero, las velocidades de transmisión deben ser la misma. así que los datos son transmitidos como caracteres individuales. Cada caracter es sincronizado por la información contenido en su bits de inicio (header, cabezal) y su pare (trailer, cola). Los tradicionales modems de conexiones Dial-Up usan transmisiones Asincrónicas.

Flujo de Transmisión de Data

Los tres métodos de operación de circuito son como sigue a continuación:

- **Simplex**
  Los datos viajan en una sola dirección, similar a un sistema de Aviso Público (Public Address, PA).
- **Half Duplex**
  Los datos viajan en dos direcciones pero solo en una dirección a la vez, similar a los walkie-talkie. El Ethernet usa transmisiones half-duplex.
- **Full Duplex**
  Los datos viajan en dos direcciones simultáneamente, similar a una conversación telefónica. Ethernet Full-duplex, es una extensión del Ethernet, soporta transmisiones full-duplex en un ambiente asistido por switch99999999999.

Transmisiones de Banda Base y Banda Ancha (Basaban y Broadband)

En redes el bandwidth (ancho de banda) es la medida de la capacidad de transmisión de un medio en particular. Esta velocidad se cuantifica como un número de bits que puede ser transmitido por segundos. El ancho de banda de un medio de transmisión puede ser dividido en canales; así que cada canal es una porción de la capacidad total disponible para transmitir data. Los dos métodos usados para colocar ancho de banda a los canales son el Basaban y broadband.

Basaban (Banda base)

Basaban usa todo el ancho de banda del medio para un único canal. Aunque es usado principalmente para señales digital, Basaban puede también conducir señales análogas. La mayoría de las LANs, como son las redes Ethernet y Token Ring, usan señales digitales Basaban.

Basaban usan un tecnología de transmisión llamada TDM (Time División Multiplexing, TDM). Esta tecnología TDM envía multiple señales sobre una sola ruta de transmisión y lo logra tejiendo la señales. Por ejemplo, tres señales (1, 2, y 3) pueden ser enviadas así 112233112233. El dispositivo que recibe separa este flujo único en las tres señales originales. TDM Estatistico99999 (StatTDM) le da más prioridad a señales más urgentes.

Broadband

Broadband divide el ancho de banda del medio en múltiples canales, y cada canal transmite una señal por separado. Este método permite que un singular medio de transmisión trasmita varias conversaciones simultáneamente y sin ninguna interferencia. El método broadband es usado exclusivamente para señales análogas; este no puede ser aplicado a señales digital porque estas señales pueden interferirse entre ellas. así que, transmisiones broadband requieren que un modem le convierta señales digitales a análoga y vice versa.

El método broadband usa un tecnología llamada FDM (Frequency División Multiplexing). Al igual que TDM, FDM también transmite multiple señales sobre una sola ruta de transmisión. Pero, cada señal en FDM transmite en un rango de frecuencia única, o carrier. Transmisiones broadband son muy comúnmente usadas para TeleCable y sistemas inalámbricos o wireless.

El término broadband es comúnmente usado para describir cualquier transmisión de data de alta velocidad que proveé servicios a velocidad de T1 (1.544 Mbps) y más alta. Eso si, las capacidades de la tecnología broadband varia inmensamente dependiendo de la situación, y velocidad actual de transmisión puede que no logren los niveles de Tl o tal ves sobre pasarlos por mucho. Por lo general, broadband implica velocidades de transmisión muy superior a aquellas que estaban disponibles en el pasado.

Topologías lógicas y Física

Topologías lógicas se refiere a la ruta real sobre una red de una señal generada. Bus y Ring son dos tipos de topologías lógicas. Una red lógica del tipo bus genera una señal a todos los dispositivos en una red. Una red lógica del tipo ring genera una señal que viaja en una dirección por una ruta determinada. En este capítulo discutiremos a más profundidad redes lógicas introduciendo sus métodos de acceso.

Topologías físicas se refiere a la manera en que los dispositivos de una red están conectados. Ya hemos vistos topologias físicas cuando estudiamos topologias bus, star, ring, y mesh, anteriormente en este mismo capítulo. Es muy importante entender la diferencia entre topologias lógicas y físicas.

DATA LINK LAYER (CAPA de ENLACE de DATOS)

En esta sección estudiamos la segunda capa de la pila OSI, la capa de Enlace de Datos. Analizamos direccionamiento, controlando y sincronizando flujo de data y revisamos errores y lo corregimos.

La capa de Enlace de Data maneja:

- **Data addressing**
  Direccionamiento es la asocia de una dirección con cada host del lado de la capa Física, similar a una dirección postal.
- **Flow control**
  Control de Flujo, controla flujo de data sobre la capa Física; Ej., ¿qué pasa si dos hosts desean enviar data simultáneamente? (Este es mejor conocido como Control de Acceso al Medio.)
- **Data integrity**
  Integridad de Data se asegura que la data llega a su destino intacta y en el orden correcto- revisa por errores.

Estos tres requerimientos son esenciales para cualquier comunicación confiable sobre un medio físico.

Los siguientes tópicos son discutidos en la siguiente sección:

- Direccionamiento de Data (Data Addressing)
- Control de Flujo (Flow Control)
- Integridad de Data (Data Integrity)
- Tramas (Frames)
- Acceso a la Capa Física

Direccionamiento de Data

Las direcciones de la capa de Enlace de Data deben ser únicas en toda la capa Física (Ej., sobre una red local). De otra forma, tráfico de la red no puede ser direccionado a su destino correcto.

Direcciones MAC

Cada dispositivo en una red debe tener una dirección única. Los usuarios pueden usar nombres como Oficina, pero la capa de Enlace de data usa completamente un sistema numérico de direcciones. Este es de 48 bits (6 bytes) de largo, con 3 bytes reservadas para el ID del fabricante y 3 para uso local. Esta direc-

ción, el número MAC (Media Access Control), es asignada por el protocolo ARP (Address Resolution Protocol). Direcciones MAC son direcciones únicas que son parte del NIC. Cada dirección es asignada por el fabricante y es usada para identificar la computadora en la red.

Direcciones MAC son llamadas direcciones físicas y no lógicas. Direcciones lógicas son encontradas en la Capa 3 (Capa de Red) y incluyen direcciones IP y IPX. Direcciones lógica son usadas para enviar data sobre el Internet y la red enviadas a destinaciones remota. Direcciones físicas son encontradas en la Capa 2 (Capa de Enlace de Data) y son muy a menudo parte de la interfase Física. Direcciones Física son usadas solamente para enviar data entre dos dispositivos en un sólo enlace de red.

Direcciones MAC usan un hexadecimal de 12 dígitos para formar una dirección de 48-bit (6 bytes). Cada mitad de la dirección es usada para un propósito diferente.

00-0A-95	**-**	**FF-FE-85**
Código del Fabricante		**número Serial del Interfase**

El código del fabricante está identificado en los primeros 24 bits (3 bytes).

El número serial de la interfase está identificado en los últimos 24 bits. Determinado por el fabricante , es siempre único para ese fabricante. En teoría, ninguno dos direcciones MAC son idénticas.

Determinar direcciones MAC en GNU/Linux

Usted puede determinar la dirección MAC de su sistema usando el comando ifconfig (interfase configuration) con la opción -a (deberá estar ingresado como root y la interfase deberá estar activa para que se pueda desplegar la dirección MAC).

Aquí un ejemplo de la salida del comando ifconfig:

ifconfig

Resultados del comando ifconfig, dependiendo de su NIC y la configuración de su red:

```
# ifconfig eth0
eth0    Link encap:Ethernet  HWaddr 00:08:C7:F3:E6:E4
        inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
        EtherTalk Phase 2 addr:65280/106
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1179092 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1583178 errors:1 dropped:0 overruns:0 carrier:1
        collisions:0
        RX bytes:208032912 (198.3 Mb)  TX bytes:1540098887 (1468.7 Mb)
```

Ejercicio 2-1: Identifique la Dirección de Hardware en GNU/Linux

En este ejercicio, usted va a localizar la dirección del hardware de su computador en GNU/Linux. No se proveen soluciones a este ejercicio.

1.- Ingrese al sistema como root.

2.- El prompt digite:

ifconfig

3.- Aparecerá la información de su IP y la información de su NIC. Observe la dirección de su hardware (HWaddr).

Address Resolution Protocol (ARP)

El protocolo de Resolución de Direcciones para mapear direcciones IP a direcciones de hardware, una variedad de opciones pueden ser usadas. ARP es usado Típicamente en redes IEEE 802 (Ethernet, Token Ring). Este proveé una vía para que un sistema difunda una petición para una dirección IP para una dirección de hardware. Los broadcast de ARP a todos los hosts en una red local. La información puede ser proveída por el host en específico o por otros hosts que tal ves ya saben la repuesta. REF 826 cubre la conversión de direcciones IP a direcciones MAC.

Aquí presentamos un ejemplo de como funciona la resolución con ARP:

- Cuando el host cliente1 necesita resolver la dirección del cliente2, este envía un broadcast de un paquete especial que pide a cliente2 que responda con su dirección Física. Este mensaje es conocido como el paquete de petición de ARP.
- Aunque todos los hosts en la red reciben la petición, sólo el nodo que reconoce su dirección de Internet responde con su dirección Física. este mesase es referido como la respuesta ARP.

Los hosts que usan ARP mantienen un cache de las direcciones adquiridas recientemente de Internet-a-Física para así ellas no tener que usar ARP repetidamente. El tiempo en averaje que una dirección ARP permanece en un cache GNU/Linux es de 20 minutos.

El comando para ver el cache del ARP en GNU/Linux es:

arp

Este comando genera el siguiente resultado:

EJEMPLO DEL COMANDO ARP 99999999999

Cabezal del ARP (Header)

El ARP header tiene una longitud de 28 bytes. El formato del header consiste de palabras de 32-bit. Este formato es usado para el propósito de desplegar para así poder entenderlo y explicarlo. El header del ARP consiste de siete palabras de 32-bit, lo cual es 28 bytes.

0		32
Hardware Type	Protocol Type	
Hardware Length	Protocol Length	Operation
Source Hardware Address		
Source Hardware Address	Source IP Address	
Source IP Address	Destination Hardware Address	
Destination Hardware Address		
Destination IP Address		

Campos de ARP Header

Un mensaje ARP es encapsulado en una trama de Ethernet. El campo frame type del header de Ethernet para los paquetes ARP se establece al hexadecimal 0806. Los siguientes son los campos del header:

Hardware Type (16 bits)
Define el tipo de dirección de Hardware (uno para Ethernet)

Protocol Type (16 bits)
Define el tipo de dirección del protocolo (0x0800 para las direcciones IP); el mismo valor que el campo tipo de frame de Ethernet.

Hardware Length (8 bits)
Tamaño, en bytes, de la dirección de hardware [el valor es de 6 bytes (48 bits) para Ethernet]

Protocol Length (8 bits)
Tamaño, en bytes, de la dirección del protocolo [el valor es de 4 bytes (32 bits) para el IP]

Operation (16 bits)

Define el tipo de ARP: 1=Peticiones de ARP 2=Respuestas de ARP

Source Hardware Address (32 bits)

Dirección de hardware del que envía (6 bytes para Ethernet)

Source Hardware Address (16 bits)

Dirección de hardware del que envía (continuado)

Source IP Address (16 bits)

Dirección del protocolo de equipo que envía (4 bytes para el IP)

Source IP Address (16 bits)

Dirección del protocolo de equipo que envía (continuado)

Destination Hardware Address (l6bits)

Dirección de hardware de equipo de destino (6 bytes para Ethernet)

Destination Hardware Address (32 bits)

Dirección de hardware de equipo de destino (continuado)

Destination IP Address (32 bits)

Dirección del protocolo de equipo que envía (4 bytes para el IP)

Ejercicio 2-2: Ver el Cache del ARP en GNU/Linux

En este ejercicio, echaremos un vistazo al cache del ARP, entonces procederemos a agregar y eliminar entradas en el cache. No se proveen soluciones para este ejercicio.

1. En el prompt de root (#), escriba el siguiente comando:
   # arp

2. Visualice las entradas del cache de ARP. Si una entrada no existe de su segundo equipo, creala ejecutando este comando en el prompt:
   # ping [dirección IP de segundo sistema no existe]

3. Presione las teclas CONTROL+C para detener el proceso de ping. Visualice de nuevo el cache de ARP, escribiendo este comando:
   # arp

   Ya debe existir una entrada para su segun sistema en el cache de ARP.

4. escriba la entra ARP de su segundo sistema. Incluya solamente las direcciones IP y de hardware:

   Dirección IP: ___________

   Dirección de HW:_________

5. Si existen entradas adicionales en su cache de ARP, intente determinar a que nodos de la red pertenecen-Por ejemplo se ha usted comunicado recientemente con otro equipo en la red?

6. Para eliminar una entrada de ARP, use la opción -d. En el prompt simplemente escriba:
   # arp -d [Dirección IP segundo equipo en el sistema]

7. Visualice el cache del ARP escribiendo:
   # arp

   Note como el ARP de su segundo sistema ya no aparece como entrada en el cache ARP.

Protocolo de Resolución de Direcciones Inverso (RARP)

El RARP (Reverse Address Resolution Protocol) es usado por los sistemas sin discos para poder encontrar su dirección de Internet en la red. El sistema sin disco (diskless system) efectúa una petición RARP a través de un broadcast, el cual proveé su dirección Física en la red. Un servidor RARP entonces procede a enviar un RARP de repuesta, normalmente un unicast, el cual específica la dirección IP de la estación

sin discos.

Ethernet Address (48-bit) =====> IP Address (32-bit)

Generalmente los sistemas sin discos dependen de RARP durante su inicialización. Soporte de RARP puede ser proveído en ROM ya que es pequeño y simple. Debe existir por lo menos un servidor RARP en la red para que RAPR funcione.

Descripción de RARP

El header RARP tiene la misma longitud como el header ARP (28 bytes) y también es encapsulado en la porción de la trama Ethernet que pertenece al campo de Data. Este permite no sólo a que las computadores adquieran su dirección de Internet, sino que también esas de los otros sistemas. Todo computador en la red recibe una petición, pero sólo el servidor RARP procesa la petición y envía la respuesta.

Los headers del RARP y el ARP son similares. Sus únicas diferencias son ilustradas a continuación:

• Los paquetes de petición y respuestas RARP tienen su encabezado de Ethernet el campo tipo de trama establecido al hexadecimal 8035.
• El campo de operación define los tipos de mensajes RARP:
3=RARP request(petición)
4=RARP reply (`respuesta)

Control de Flujo (Flow Control)

Si A envía datos a B más rápido que B puede procesarlo, B se sobrecarga. El Control de flujo se implementa estableciendo el número máximo de tramas que A puede enviarle a la ves a A antes de que B le permita continuar.

Estas medidas son implementadas por lo general en las capas superior de los protocolos pero también puede ser implementado en la capa de Enlace de Data. En el caso del Ethernet y el TCP/IP, la capa de Transporte (TCP) maneja el control de flujo donde sea necesario aplicarlo. Hay un mecanismo para manejar la sobrecarga de los hosts llamado Source Quench. Este mecanismo yace entre el ICMP (Internet Control Message Protocol), un protocolo de la capa de Red que maneja el control de flujo para el IP. El mecanismo envía un mensaje al dispositivo que envía (origen) para detectar su velocidad de envío. Este es reactivo y no premeditado. Hay una gran variedad de mensajes ICMP, así como es determinar si un host esta despierto o no, petición de eco (echo), parámetro equivocado en un header de IP, o redireccionar mensajes por otra ruta.

El protocolo de alto nivel HDLC (High-level Data Link Control) es un ejemplo de control de flujo en esta capa. Es usado principalmente por paquetes provenientes de sistemas que proveen switching a paquetes generados por el teléfonos como son los protocolos X.25 o el Point-to-Point Protocol (el PPP, será discutido más adelante).

Integridad de Data

La integridad de data es proveída con la división de la data en tramas y agregandole un CRC (Cyclic Redundancy Check). El CRC es un código computacional diseñado matemáticamente para servir como una firma electrónica para trama de la data involucrada (Header + Data).

Data Frame ---> | HEADER | DATA | CRC |

Si aunque sea un sólo bit de data dentro de la trama se corrompiese dentro del frame (por ejemplo, interferencia eléctrica), entonces cuando la data es recibida en el destino un nuevo CRC es computado desde la trama de datos recibida, el nuevo CRC variará comparado con el transmitido originalmente en la trama. así que, el host de destino sabrá que la trama está corrompida y entonces decidirá que otra acción tomar. En

muchas tecnologías LAN, la acción simplemente es dejar caer la trama y olvidarse que nunca existió sin notificar al enviante o originador.

La matemática detrás de los CRCs ha sido cuidadosamente diseñada para reducir el chance de que tramas dañadas generen el mismo CRC que el original sea virtualmente cero. así que, previsto que el CRC calculado una vez recibido el frame iguala el enviado, entonces se asume que la trama está intacta.

Timers y Timeouts

El revisado del CRC se asegura que tramas o frames corrompid sean aceptadas por el host de destino. Pero, si es necesario que una trama corrompida sea devuelta al host de origen. ¿Como se lleva esto acabo?

El host A establece un tiempo al momento de enviar la trama. Si el, el host A, no recibe una confirmación (acknowledgment) a tiempo, la trama es reenviada y el timer es reiniciado.

Mientras que esto es una característica de la capa de Enlace de Data, no es implementado allí (no en redes de Ethernet), y esta responsabilidad de la verificación del transporte de los datos es dejado a capas más altas del OSI. Con el TCP/IP sobre Ethernet, la capa de Red (con el IP, también sucede lo mismo veremos después) es quien maneja esto.

Este proceso requiere que cada trama sea enumerada para poder llevar acabo su confirmaciones y que sus tramas reconocidas puedan ser identificadas.

Tramas/Frames

Es necesario que los datos que colocamos en un alambre sean particionados en pequeños paquetes discretos de información. Estos paquetes son denominados tramas en la capa de Enlace de Data. Existe un gran número de tramas que podemos usar que varían por su formato, tamaño, codificación y todos los parámetros del protocolo que pueden ser usados. Diseñar un nuevo protocolo de Enlace de Data no es una tarea fácil! Aquí mencionamos algunas de las cosas que se deben tomar en cuenta:

- ¿Cómo son las tramas delimitadas?
- ¿Qué tamaño deben ser las tramas?
- ¿Cómo deben ser los bits codificados en el medio físico?
- ¿Qué sucede si dos computadoras intentan enviar data a la misma vez?
- ¿Cómo sabe usted si la trama llegó a su destino?
- ¿Cómo controla usted a un host de no sobrecargar el otro con el envío de su datos?
- ¿Cómo podemos optimizar todo este proceso?

Hay tres maneras diferentes de marcar el la longitud de una trama y como mantenerlas separadas. Podemos marcar el final de una trama con delimitador especial. También podemos colocar el largo de la trama en el cabezal (header) de la trama misma. Otra posibilidad es que todas las tramas sean de la misma longitud y enviarlas de acuerdo a un mecanismo especial de el timing. La mayoría de medio físicos usan el método de colocar la longitud de sus tramas en el header.

End marking: | START | HEADER | DATA | CRC | END| ----*Marcadas al final*
Length mark: | HEADER | LENGTH | DATA | CRC | ---*Marcadas su longitud*
Timing: | HEADER | DATA | CRC | ......pausa después de cada trama. ---*Un timer interno*

Un preámbulo es generalmente agregado al principio de cada trama para permitirle a la tarjeta de interfase que sincronice su reloj con la trama que llegan

.

| 101010010...01001001...010101010 | TRAMA |

Tamaño de la Trama

El tamaño máximo de un frame en una red, determina la carga máxima de datos que cada frame puede cargar a la vez. Esta carga máxima es conocida como el MTU (Maximum Transmission Unit) del sistema. Si un host desea enviar pedazos de data más grande que el MTU, esta deberá repartirse en más de un trama. Consideremos lo siguiente:

- ¿De que tamaño debe ser la trama?
- Los headers y los CRCs son por lo general de un tamaño fijo.
- Tramas con tamaño variables son usadas con un tamaño permisible máximo o mínimo.

Codificación Manchester

Hay mucha maneras de codificar una cadena de datos a voltajes eléctricos en una línea serial. Una manera común es establecer cada estación en una frecuencia en particular, digamos 9600 baudios, y traducimos un bit 0 a 0 voltios y el bit 1 al voltio de la línea. Entonces rodeamos cada byte con un start, stop, y bits de paridad para sincronizar y revisar la información. En la codificación Manchester, usamos la transición de voltaje y no el voltaje simple. Otro método es la codificación Manchester diferencial que consiste en que un bit con valor 1 se indica por la ausencia de transición al inicio del intervalo, y un bit con valor cero se indica por la presencia de una transición al inicio del intervalo. En ambos casos, existe una transición en la parte media del intervalo.



Acceso a la Capa Física

Necesitamos un método para controlar el acceso a la capa Física. Si todos los ordenadores intentan enviar datos a la vez, se perderían en colisiones. Existen dos maneras esenciales de enfrentar este dilema: Carrier Sense Multiple Access/Collision Detection (CSMA/CD) y paso de ficha (token passing).

Las colisiones resultan en perdida de data. Los CRCs no se ajustan a la reconstrucción de data dañado o perdida en una trama, así que no se efectúan intentos de salvar los resultados de una colisión. Las colisiones pueden ser detectadas fácilmente, pero, por host que continuamente monitorean el medio por otras data mientras sigue enviando.

CSMA/CD

El protocolo CSMA-CD funciona de la siguiente manera: un nodo que desea transmitir espera a que el canal esté aislado, escuchando por períodos de silencio antes de enviar sus tramas (CS), una vez que se encuentra en este estado empieza la transmisión. Las tramas toman un tiempo determinado para viajar por el medio. Si el nodo A empezara también a transmitir mientras B aún piensa que la línea está desocupada, en este instante se produciría colisión, por lo tanto se detiene la transmisión y se retransmite tras un retraso aleatorio.

Esto es muy similar a conversaciones de dos o más personas que se interrumpen comenzando a hablar a la misma vez. Dos computadores pueden determinar en un mismo momento que el cable esta libre.

Todo esto es manejado con el CSMA/CD, de esta manera:

- Los Hosts pueden detectar cuando una colisión está ocurriendo y parar de enviar (CD).
- Ellos entonces esperan un tiempo al azar antes de intentar enviar de nuevo si la línea ya esta limpia.
- Reintentos futuros pueden ser colocados más atrás aún en tiempo exponencial

Ejemplo es si se falla la segunda vez es dos veces más, entonces 4 veces más, 8 veces, 16 veces, etc.

Las variaciones de CSMA/CD son denotadas por su persistencia, esto significa que, aunque ellos traten siempre de enviar inmediatamente después que la línea este limpia.

CSMA 1-persistente

El protocolo CSMA 1-persistente funciona de la siguiente forma: cuando tiene que transmitir un frame, primero escucha el canal y si está libre envía el frame, caso contrario, espera a que se libere y en ese momento lo envía. Se denomina CSMA 1-persistente porque existe la probabilidad 1, es decir, certeza de que el frame se transmitirá cuando el canal esté libre. En una situación real con alto tráfico es muy posible que cuando un nodo termine de transmitir existan varios esperando enviar sus datos, y con CSMA 1-persistente todas los frames serán emitidos a la vez y colisionarán, pudiéndose repetir el proceso varias veces con la consiguiente degradación del rendimiento. Cabe señalar que una colisión ocurrirá aunque no empiecen a transmitir exactamente a la vez, basta simplemente con que dos nodos empiecen a transmitir con una diferencia de tiempos menor que la distancia que los separa, ya que en tal caso ambos detectarán el canal libre en el momento de iniciar la transmisión. Se deduce entonces, que en este tipo de redes el retardo de propagación de la señal puede tener un efecto importante en el rendimiento. El rendimiento obtenido con este protocolo puede llegar al 55% con un grado de ocupación del 100%.

El Ethernet usa el CSMA/CD 1-persistente, significando que esta siempre trata de enviar las tramas a respuesta de una línea vacía (esto significa que, la probabilidad de transmisión es 1).

CSMA: no persistente

Corresponde a una modificación del protocolo anterior que funciona de la siguiente manera: antes de enviar se escucha el canal, si el canal está libre se transmite el frame. Si está ocupado, en vez de quedar escuchando, se espera un tiempo aleatorio, que viene dado por un algoritmo llamado de backoff, después del cual se repite el proceso. El protocolo tiene una menor eficiencia que CSMA 1-persistente para tráficos moderados, pues introduce una mayor lactancia; sin embargo se comporta mejor en situaciones de tráfico intenso ya que evita las colisiones producidas por las estaciones que se encuentran a la espera de que termine la transmisión de un frame en un momento dado.

CSMA p-persistente

Este protocolo utiliza intervalos de tiempo y funciona de la siguiente manera: cuando el nodo tiene algo que enviar primero escucha el canal, si está ocupado espera un tiempo aleatorio. Cuando el canal está libre se selecciona un número aleatorio con distribución uniforme entre 0 y 1, si el número seleccionado es menor que p el frame es transmitido. En caso contrario, se espera el siguiente slot de tiempo para transmitir y repite el algoritmo hasta que el frame es transmitido o bien otro nodo utiliza el canal, en cuyo caso se espera un tiempo aleatorio y empieza de nuevo el proceso desde el principio. La eficiencia del protocolo es, en general, superior a la de CSMA 1-persistente y CSMA no persistente.

Entre las Ventajas CSMA/CD se incluyen:
- Simplicidad del algoritmo
- Fácil incluir y eliminar nuevos hosts
- Implementación sencilla, bajo coste y fiabilidad
- Requiere poca administración
- Técnica suficientemente probada
- Buen manejo de carga de variables
- Buen rendimiento hasta determinado nivel de carga (aproximadamente 85%)

Inconvenientes CSMA/CD incluyen:
- Técnica LIFO ante colisiones
- Longitud de los mensajes mínima

• Difícil distinción entre ruido y colisiones
• La atenuación complica la detección de colisión
• No permite la gestión de prioridades
• No garantiza tiempo de entrega
• Difícil ejecutarse a alta velocidad
• Rendimiento pobre conforme aumenta la carga (colisiones masivas y retrasos)

Aún con estas desventajas, el CSMA/CD ayuda a hacer que el Ethernet sea la elección más popular en networks al día de hoy causado por su simplicidad de implementación y el bajo coste de su hardware.

Token Passing

Este protocolo, que se utiliza en redes Arcnet y Token Ring, se basa en un esquema libre de colisiones, dado que la señal (token) se pasa de un nodo o estación al siguiente nodo en la anillo. Con esto se garantiza que cada estación solamente envía tramas mientras posé el token y que un sólo paquete viajará a la vez en la red lo que significa que no habrán colisiones.

Claro está, no colisiones no significa que no hay problemas. Por ejemplo, ¿que sucede si se corrompe el token mientras se pasa alrededor del ring?

Cada host tiene que saber cual máquina están en cada lado de su anillo lógico para así poder aceptar y pasar el token. Considere que sucede si agregamos un host extra a la red- no esta dentro del anillo y por esto no recibirán automáticamente el token.

Resolver este y otros problemas hacen de pasar el token (aunque simple a primera vista) un sistema mucho más complejo de implementar que CSMA/CD.

La configuración anterior es conocida como token bus. La de Token Ring es muy similar, pero los hosts son ambos lógicos y físicamente conectados al ring.

Entre las ventajas de token passing se incluyen:
• Buen manejo de carga variable
• Sin Colisiones
• Garantía de tiempo de entrega/mínimo uso ancho de banda
• Maneja bien sobrecarga
• Puede ser ejecute a alta velocidad

Entre las desventajas de token passing se incluyen:
• Agregar y eliminar hosts no es simple tarea.
• Si un host falla, la red se apaga.
• En practica, es compleja de implementar y administrar.
• Por lo general, altamente compleja significa alto coste.

Un sistema token-passing es más eficaz en un ambiente donde la data debe tener una garantía de tiempo de entrega o una disponibilidad de ancho de banda mínimo. Cuando el nivel de ejecución es lo más importante, o la red es probable que este muy sobre cargada, el token passing es lo ideal.

ESTÁNDARES DE COMUNICACIÓN

Un factor que distingue las tecnologías LAN es sus métodos de acceso. Los métodos de acceso se refiere a la manera que los datos son colocados en el alambre físico. La serie 802 incluye CSMA/CD, token, y el método de acceso de prioridad en demanda (demand priority access methods). Cada uno de estos

métodos será discutido en esta sección.

En esta sección, los siguientes tópicos serán discutidos:
• Estándares LAN
• Estándares WAN

Comparativa de Tecnologias LAN					
	10Base2 Ethernet	**10Base5 Ethernet**	**10BaseT Ethernet**	**100BaseT Ethernet**	**IBM Token Ring**
Medium	Coaxial	Thick Coaxial	Twisted Pair	Twisted Pair/Fiber	Twisted Pair/Fiber
Topology	Bus	Bus	Star	Star	Ring
Max segments length	185m	500m	108m	100m	Max 260 hosts
Max nodes/segment	30	100	1	1	N/A
Connectors	BNC	50/8-pin AUI	RJ-45	RJ-45/Photodiode	9-pin/RJ-45
Speed	10Mbps	10Mbps	10Mbps	100Mbps	4 o 16 Mbps

• Serial Line Internet Protocol (SLIP)
• Protocolo de Punto-a-Punto (Point-to-Point Protocol, PPP)
• Redes Digital de Servicios Integrados (Integrated Services Digital Network (ISDN)
• Cable Modem
• Digital Subscriber Line (DSL)
• X.25
• Conmutación rápida de paquetes/Fast Packet Switching
• Sistemas de T-Carrier
• Sistema de E-Carrier

Estándares LAN

El Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers, IEEE) es una organización de profesionales que se encarga de crear estándares para computadores y las comunicaciones. La serie de Estándares IEEE 802 especifican varias tecnologías LAN. Refierase a las páginas web de la IEEE en http://www.ieee.org para más información.

En el ambiente TCP/I Ethernet es el LAN más comúnmente usado. La arquitectura lógica es de muchos sistemas diferentes conectados a un cable central o bus, llamado el ether. De esta manera, cada nodo puede potencialmente conectarse con cualquier otro nodo.

Otra tecnología dominante para LANs basadas en arquitecturas de PC es el Token Ring, más trabajado por IBM. Se fundamenta en que un token circula alrededor de todo el añillo todo el tiempo. Si una estación de trabajo desea enviar data, deberá adquirir el token y enviar su mensaje adjunto a este. Esto se asegura de que sólo una estación transmitirá mensajes a la vez.

Aquí le presentamos una comparativa de varias tecnologías LAN comunes.
Esta sección cubre los siguientes Estándares de red IEEE 802:
• IEEE 802.2
• IEEE 802.3-Ethernet
• IEEE 802.3u-Fast Ethernet

- IEEE 802.3z y el 802.3ab-Gigabit Ethernet
- IEEE 802.5-Token Ring
- IEEE 802.12-100VG-AnyLAN

En esta sección discutiremos dos Estándares de LAN que no están incluidos en los Estándares IEEE. Estos Estándares son o propietarios o estandarizados por otra organización. Ellos son:
- Apple LocalTalk
- Fiber Distributed Data Interfase (FDDI)

IEEE 802.2

Todos los Estándares en la serie IEEE 802 usan el estándar 802.2. Este estándar también es usado por redes de Fiber Distributed Data Interfase (Interfase de Data Distribuida en Fibra, FDDI), el cual no es un estándar de la IEEE. El estándar 802.2 divide la capa de Enlace de Data del modelo OSI en dos subcapas: Control de Enlace lógico (Logical Link Control, LLC) y Control de Acceso al Medio (Media Access Control, MAC).

La capa de la subcapa LLC proveé servicios orientados a conexiones y servicios sin conexión en la capa de Enlace de Data, cual administra transmisiones y puede proveer control de flujo.

La subcapa de MAC proveé:
- Acceso al medio del LAN. La MAC es responsable por colocar la data en el alambre.
- La dirección MAC (también llamada la dirección de hardware, dirección física o dirección de Ethernet).

Aunque Ethernet no usa técnicamente el estándar 802.2, este es compatible con el y comparte varios elementos importantes, como es la subcapa MAC.

IEEE 802.3-Ethernet

Ethernet es sin lugar a duda una de las tecnologías LAN más exitosas y es un predecesor al estándar IEEE 802.3. Es un sistema de transmisión para la comunicación entre sistemas. Este utiliza los Estándares de alambrado 10Base2, 10Base5, o 10BaseT. También puede usar cables de fibra óptica.

Aunque Ethernet y IEEE 802.3 son ambos soportados y usados intercambiables, Ethernet no conforma totalmente con el estándar 802.2.

La diferencia entre IEEE 802.3 y Ethernet no afecta a los fabricantes de hardware porque ambos el IEEE 802.3 y el Ethernet soportan direcciones MAC y la misma capa Física. Además, existe software que diferencian entre las subcapas.

Todas las redes que usan Ethernet/IEEE 802.3 (incluyendo IEEE 802.3u, 802.3z, y el 802.3ab) usan CSMA/CD. Una estación de trabajo debe asegurarse que ninguna otra comunicación se encuentra ya desarrollandose. Si ninguna otra estación se encuentra transmitiendo, el que envía puede empezar de inmediato. Las colisiones ocurren cuando dos o más estaciones determinan que el canal esta libre y empiezan a transmitir simultáneamente. En caso de una colisión, toda la transmisión cesa, mientras las estaciones en conflictos son notificadas. Las estaciones que reportaron colisiones entonces esperan un tiempo al azar antes de transmitir de nuevo.

Las transmisiones son locuciones a todas las estaciones de trabajo. Solamente el sistema de destino responde; todos los otros sistemas descartan la transmisión. Este proceso puede crear tráfico pesado en una red. así que , es importante dividir grandes redes de ethernet en segmentos (usando un bridge).

IEEE 802.3u- fast Ethernet

El Fast Ethernet es una versión más rápida del IEEE 802.3. Fué desarrollada originalmente por fabricantes como son 3Com, Intel, Cabletron, SynOptics, Digital, Grand Junction Networks. El comité del IEEE 802.3 es responsable por el Fast Ethernet. El objetivo principal del estándar Fast Ethernet es promover el uso del Ethernet de 100 Mbps usando el mismo método de acceso, CSMA/CD.

Fast Ethernet soporta los Estándares de cableado de 100BaseTX y 100BaseT4, cuales requieren cablea-

Descripción	Ethernet	Fast Ethernet
Speed	10 Mbps	100 Mbps
IEEE Standard	IEEE 802.3	IEEE 802.3u
Access Method	CSMA/CD	CSMA/CD
Topology	Bus/Star	Star
Cable Support	Coax/twisted Pair/fiber	Twisted Pair/Fiber
UTP Link Distance (Maximum)	100 meters;	100 meters;

do UTP Categoría 5 para soportar los 100 Mbps. Este puede también usar 100BaseFX, cual es el cableado de fibra óptica. Fabricantes soportan tarjetas nics Fast Ethernet que usan ambas velocidades de transferencia 10 Mbps y 100 Mbps.

La mayoría de administradores de redes ya se encuentran actualizados a desde las 10BaseT a las 100BaseTX o 100BaseT4. En muchos otros casos encontraras que esta actualización fué efectuada con el reemplazo de las NICs 10BaseT con las 100BaseTX o las NICs 100BaseT4 y actualizando los HUBS para soportar ambas la 10BaseT y 100BaseTX o las 100BaseT4. Este proceso puede ser menos costoso que el de actualizar a una NIC 100BaseFX, Token Ring de 16 Mbps, 100VG-AnyLAN, o redes FDDI.

En la siguiente tabla mostramos las diferencias entre Ethernet y el Fast Ethernet:

IEEE 802.3z y 802.3 ab-Gigabit Ethernet

El Gigabit Ethernet es la tecnología LAN más veloz disponible 802.3. Es usada principalmente en los backbones de las redes corporativas. El estándar de Gigabit Ethernet transfiere data a velocidades hasta los 1,000 Mbps usando el método de acceso CSMA/CD.

Los dos tipos de Gigabit Ethernet son IEEE 802.3z y 802.3ab. El estándar 802.3z es especificado para cables especiales de cobre y cables de fibra óptica. El estándar 802.3ab específica el Gigabit Ethernet sobre cable UTP Categoría 5 UTP (hoy día ya se llego a la Categoría 6 de Cables UTP).

Gigabit Ethernet soporta el estándar de cableado de 1000BaseT, el cual usa cableado UTP Categoría 5 para poder soportar los 1,000 Mbps. Este también puede usar 1000BaseCX, 1000BaseSX, y 1000BaseLX , cual usa cableado de fibra óptica.

IEEE 802.5-Token Ring

Las red Token Ring está especificada en la definición IEEE 802.5. Token Ring fué originalmente desarrollada por IBM para su ambiente de mainframe, y el estándar IEEE 802.5 cumple con el desarrollo original de la corporación.

Mientras que Ethernet usa el método de acceso CSMA/CD. Redes Token Ring usan el método de acceso de token-passing (paso de ficha) que discutimos anteriormente. En vez enviar broadcasts como hace el Ethernet, una red Token Ring pasa un token en una dirección alrededor de la red. Cada nodo procesa el token

para determinar su destino. El después de revisar o lo acepta o lo coloca en el anillo de la red. Uno más tokens pueden circular el anillo (ring). Con el paso de ficha, no ocurren colisiones; es muy similar a calles de una sola vía sin esquinas que la crucen.

El estándar IEEE no específica un estándar de cableado, pero las redes de Token Ring de IBM usan cables twisted-pair. velocidad de transferencia de 4 o 16 Mbps son posible con redes Token Ring. Redes Token Ring aparentan estar usando topología de red STAR cuando en realidad usan un dispositivo algo parecido a un HUB llamado MSAU (Multistation Access Unit) para formar el anillo.

El MSAU crea el anillo usando conexiones internas.

El fallo de un nodo en la topología anilla RING puede causar la caída de la red por completo. así que, los MSAUs pueden identificar y sobre volar nodos fallidos en un segmento de red y continuar funcionando.

IEEE 802.12-100VG-AnyLAN

La 100VG-AnyLAN fué desarrollada originalmente por AT&T y HP y es administrada por el estándar de la IEEE 802.12. El estándar IEEE 802.12 es generalmente referido por su estándar de cableado, 100VG-AnyLAN.

Un punto clave de 100VG-AnyLAN es que esta no soporta CSMA/CD. Pero por otro lado si soporta un método de acceso llamado demand priority (prioridad en demanda) en la cual el HUB administra cuando y como el sistema puede accesar la red. Este soporta diferente niveles de prioridad, así puede garantizar que tareas con prioridad de tiempo reciban el acceso que ellas necesitan para un optimo rendimiento.

Este esquema permite poco o mejor dicho ningún espacio de contención ya que es el HUB quien determina cual estación de trabajo recibe acceso a la red. Cuando multiple transmisiones llegan al HUB simultaneas, la transmisión con la prioridad más alta es servida primero. Si la peticiones de las dos estaciones tienen el mismo nivel de prioridad, entonces se sirven ambas en corridas alternativas.

El método de acceso de prioridad en demanda usado por 100VG-AnyLAN puede ser ventajoso para algunos clientes porque este proveé una manera de garantizar prioridad para cierto tipo de tráfico en la LAN. Además, transmisiones enviadas a través del HUB no son retransmitidas (broadcast) a otras estaciones, así eliminando cualquier posibilidad de que alguien este escuchando (sniffing) nuestras transmisiones.

100VG-AnyLAN puede usar una gran variedad de cables para su estándar de alambrado (de aquí proviene el término AnyLAN). Por ejemplo, este puede usar las Categorías de UTP 3, 4, o 5 UTP, STP, o cables de fibra óptica.

Apple LocalTalk

LocalTalk es un tipo de red usado por Apple. Aunque no es un estándar IEEE, LocalTalk es importante ya que este usa un método de acceso a la red llamado Carrier Sense Multiple Access/Collision Avoidance (CSMA/GA). CSMA/CA específica que cada nodo debe informarle a los otros nodos de su intensiones de transmitir. Una vez los otros nodos han sido notificados, la información entonces es transmitida. Este acuerdo previene colisiones porque todos los nodos están al tanto de las transmisiones antes de que estas ocurran.

Ahora para resumir los métodos de acceso, IEEE 802.3 y Ethernet usan CSMA/CD, redes LocalTalk usan CSMA/GA, redes Token Ring usan un método basado en token, y las redes 100VG-AnyLAN usan prioridad en demanda (demand priority).

Fiber Distributed Data Interfase (FDDI)

FDDI es un estándar de LAN de alta velocidad. Fué desarrollado por el comité acreditador de Estándares X3T9.5 American National Standards Institute (ANSI). Al igual que el estándar IEEE 802.5 Token Ring, FDDI es basado en token.

Este estándar específica la subcapa de la MAC de la capa de Enlace de Data así como la capa Física para una LAN de 100-Mbps contra el reloj (sentido de como viaja la data), Token Ring, fibra óptica.

FDDI usa dos anillos que rotan la información para así proveer redundancia y permitir la red seguir funcionando si falla uno de los anillos. FDDI funciona bien sobre distancias sobre los 200 km (con un sólo anillo) con hasta 1,000 estaciones conectadas. En topologías de anillos dual, cada anillo se limita a solo 100 km. Como son usada para cubrir áreas metropolitanas o áreas específicas, una red FDDI puede ser clasificada

Velocidad WAN	Medio	Protocolo(s)	Rango
155Mbps+	Fibra Optica	ATM	˜10 a 100 km
1.5 a 45 Mbps+	Fibra Optica	Leased T/E 1-3	˜10 a 100 km
64 Kbps a 2+ Mbps	Fibra Optica	X.25	˜10 a 100 km
64 Kbps a 2 Mbps	Twisted pair	ISDN-30 (PRI)	Telefonica
64 Kbps a 2+ Mbps	Twisted pair	Leased Frame Relay	Telefonica
64 a 128 Kbps	Twisted pair	ISDN-2 (BRI)	Telefonica
4 a 56 Kbps	Twisted pair	Modem	Telefonica
1 a 4* bps	Aire	Semaphone	˜1 km*

como una MAN, Red de Area Metropolitana (Metropolitan Area Network).

Redes FDDI soportan ambos tipos de tráfico sincrónico y el asincrónico, y lo discutiremos más adelante.

Estándares WAN

En estos últimos 20 años, la tecnología WAN ha progresado causa de que las compañías de telecomunicaciones han invertido en aumentar sus ancho de banda. Estas tecnologías incluyen el X.25, fast packet switching, Frame Relay, y ATM. Puede ser que usted ya este familiarizado con PPP, SLIP, y ISDN, los cuales también son métodos WAN.

* Son estimados, basados en la transmisión de data binaria, condiciones climáticas adecuadas.

Recuerde que el semaphore sufre de de perdidas de bits catastróficas después de la caída del sol. La distancia son aproximaciones ya que estas dependen de los materiales usados y las condiciones en las cuales ellas operan (Ej., niveles de interferencias). Para extender el rango de estas tecnologías, se usan repetidores de señales.

Una conexión de dial-up basada en modem difiere a las soluciones de conexiones, y mientras que el equipo necesario es substancialmente más económico y fácil de instalar, el intercambio es perdida de velocidad de acceso. La gran mayoría de los modems en la actualidad tansmiten desde los 14,400, 28,800 y 56,000 bits por segundos.

Serial Line Internet Protocol (Protocolo de Internet de Línea Serial)

SLIP es uno de los primeros protocolos que fué desarrollado para enviar IP sobre líneas seriales como son los modems. SLIP es un simple conjunto de reglas que describen como los datagramas de IP pueden ser tramado para su transmisión sobre una línea serial.

SLIP fué la primera técnica desarrollada para implementar TCP/IP sobre línea serial pero nunca se formalizó como un estándar. La descripción definitiva puede se encuentra en el RFC 1055, “A Nonstandard for the Transmission of IP Datagrams over Serial Lines: SLIP.”

Algunas características de SLIP incluyen:
- Dependiente de MTU en requerimientos de enlace, Ej., tiempo de respuesta
- SD/ED caracter de escape (0xC0) (0xDB) si se encuentra presente en la data
- No revisa por error, reconocimiento (acknowledgment)
- Enlace asíncrono (debidamente, lento) solamente
- END mapeado a DB DC; ESC mapeado a DB DD

Básicamente, SLIP define como el inicio (start) y fin (end) de un datagrama de IP puede ser detectado por software leyendo bytes desde una conexión serial. Las reglas para encapsular datagramas de IP son simple y basadas en el uso de sólo dos caracteres de control:
- El caracter END (un byte único con el valor de 0xC0, o decimal 192) es usado para marcar el fin del datagrama. En la mayoría de los casos, la implementación SLIP usará este caracter para marcar el caracter de inicio (start) del datagrama aunque no es requerido.

- Para esconder el caracter de control usado por SLIP cuando ellos ocurren dentro del datagrama, el caracter de ESC (un byte único con valor de 0xDB, decimal 219) es usado. El caracter de END es transformado en una secuencia de dos bytes DB DC, y el caracter mismo de ESC es transformado en una secuencia de 2-byte DB DD.

El uso de SLIP ya no es lo que era antes, ha sido reemplazado en casi todas las plazas por el protocolo PPP.

Limitaciones de SLIP

Las limitaciones de SLIPs provienen principalmente de su simplicidad. La técnica básica de crear las tramas no permite para que más de un tipo de trama sea transmitida a la vez. Ethernet, por ejemplo, incluye un campo de identificación en el header de la trama. Esto significa que SLIP no puede compartir el enlace serial con más de una capa del protocolo de Red.

En un ambiente dial-up, es muy a menudo deseado no te tener un mapeo fijo de las direcciones IP al sistema actual. Pero, el protocolo SLIP no soporta ninguna idea de asignación de direcciones dinámicas; ambas puntas del enlace deben saber las direcciones IP entre si antes y durante la conexión.

No existe una contingencia para la detección ni corrección de errores, así que SLIP no funciona bien sobre líneas seriales ruidosas. Se asume que los protocolos del más alto nivel ejecutarán cualquier procesamiento de corrección de error necesario.

La variación de SLIP conocida como SLIP Comprimido (CSLIP) que incorpora la compresión Van Jacobson, el cual reconoce el hecho que la mayoría de header del IP es duplicado en cada paquete en todas las transmisiones. La compresión permite que los headers sean codificados en una punta del enlace y decodificada en el otro y así Típicamente lograr que sólo se envíen 8 bytes en vez de enviar 40.

Protocolo de Punto-a-Punto (Point-to-Point Protocol, PPP)

El PPP es una implementación más moderna que SLIP. Esta fué creada para sobrepasar muchas limitaciones del protocolo SLIP, y sus características pueden ser vistas como un superconjunto de las características de SLIP. Diferente de SLIP, es un estándar definido en el RFC 1548. Algunas de las características más importante de PPP son:
- Revisión Error
- Sincrónica/ Asincrónica
- Proveé autentificación débil (PAP) y fuerte (CHAP)

- Tres componentes por conexión:
  - Mecanismo de Encapsulación
  - Link Control Protocol (Protocolo de Control de Enlace, LCP)
  - Network Control Protocol (Protocolo de Control de Red, NCP)

PPP es un estándar basado en el derivada de la telefónica HDLC. El Protocolo LCP es usado para establecer, configurar, y probar la conexión data-link. Entonces el NCP es usado para negociar el Protocolo de la capa de red que será ejecutado sobre el PPP.

Por ejemplo, mientras se establece el IP, la dirección IP para cada host del enlace puede ser negociada. Hay mucho más además de esto para lo que es la especificación del PPP- variado esquemas de password/contraseñas, es sólo un ejemplo. Como PPP puede ser ejecutado virtualmente a cualquier velocidad que se requiera (enlace sincrónicos), esta es una escojencia muy generalizada para ambas conexiones punto-a-punto, la línea de modems telefónicas simple o las más rápidas enlaces de Internet (Ej., ISDN).

El PPP está en realidad compuesto por tres componentes:

- **Un mecanismo de encapsulación para los datagramas que son transmitidos**
  Este mecanismo esta basado en el protocolo HDLC pero incluye un campo adicional que identifica el protocolo que está usando el enlace. Esto permite que más de un protocolo de red comparte este enlace.

- **El Protocolo de Control de Enlace (Link Control Protocol)**
  Este protocolo se encarga de la configuración inicial (setup) y la operación del enlace a través de la negociación de varias opciones que toman lugar durante el establecimiento del enlace.
- **El Protocolo de Control de Red (Network Control Protocol)**
  Este protocolo se concierne con la operación de configurar el protocolo de red y varia dependiendo del protocolo que este en uso.

La encapsulación de PPP tiene los siguientes campos:

Flag	Byte 0x7E, usada al principio y fin de la trama
Addr	Byte 0xFF, siempre igual
Ctrl	Byte 0x03, siempre igual
Protocolo	Varia dependiendo del contenido, valores típico: 0x0021 datagramas IP, 0xC021 LCP y 0x8021 NCP.
CRC	Checksum

Configuración de Enlace en PPP

El LCP administra el estado de los enlaces:

- **Unidad de Recepción Máxima/Maximum Receive Unit**
  Equivalente al MTU para las interfases tradicionales
- **Mapeo de Caracteres de Control (Control-character mapping)**
  Este específica cuales caracteres de control ASCII deben ser mapeados a secuencias de escape.
- **Protocolo de Autentificación/Authentication Protocol**
  Si existe cualquier protocolo de autentificación, este permite que los enlaces sean revisados mientras son establecidos, generalmente basado en alguna forma de mecanismo de contraseñas.
- **La Compresión de Cierto campos en la cabecera de la Trama/Compression of certain fields in the frame header**
  Los enlaces PPP son altamente configurable. La configuración puede ser aplicada en dos etapas:
- **Durante el establecimiento del enlace, el LCP permite configuración por negociación entre los dos puntos de:**
  - Unidad Máxima de Recepción
  - Mapeo de Control de Caracter
  - Protocolo de Autentificación
  - Compresión de protocolo o dirección y campos de control
- **El NCP permite que aspectos de las características del protocolo de red del enlace a ser configurado. Para enlaces IP, es**

controlado usando el Protocolo de Control del Protocolo de Internet (IPCP). Esto permite la configuración dinámica de:

• Dirección IP

No es necesario saber la dirección IP de las dos puntas del enlace antes de establecerlo. Esto permite un uso mucho más flexible.

• Protocolo de Compresión IP

Muy a menudo usado para optimizar ejecución o rendimiento u es en particular importante en enlaces sobre seriales donde las velocidades de transmisión son más lentas que en las LANs típicas. El protocolo de compresión más usado comúnmente para TCP/IP es Van Jacobson Compression, el cual puede reducir los 40 bytes de los headers del IP y el TCP a niveles de 5 y 8 bytes.

• WAN y otras tecnologías. Los pasos básicos para establecer PPP son:

1.- Usted necesita un script para marcar el modem y luego ingresar su cuenta de usuario y contraseña.
2.- Pasar control al daemon del protocolo PPP.
3.- El daemon captura la dirección IP procedente del Proveedor de Servicios de Internet (ISP).
4.- Asociar puertos lógicos a puertos seriales.
5.- Establece a ifconfig.
6.- Establece el nombre de la interfase en la tabla de enrutamiento como la tura por defecto.

Red Digital de Servicios Integrados/Integrated Services Digital Network (ISDN)

La tarea de un modem es tomar las señales digital que se originan desde el computador y convertir los flujos de data a análogas para transmitirla por las vía de intercambio telefónico, solo para ser reconvertidas a señales digital. ISDN funciona en esencia con el reemplazo de señales análogas a soluciones digital. La conexión es ya en esta etapa de punta a punta digital y puede soportar velocidades de transferencia de data que empiezan en los 64 Kbps. Debiéramos vernos en la necesidad de mayor velocidad, podemos usar múltiples conexiones ISDN en forma multiplexadas para obtener mayor velocidad.

Existen dos tipos de interfases usuarios ISDN, estas son Basic Rate interfase (BRI/ISDN2) y la de mayor ancho de banda Primary Rate interfase (PRI/ISDN30). Cada una de estas interfases le permite a la línea establecer un número diferente de canales. Existe también dos tipos de canales en específicos; estos son canales bearer (B), el cual escucha la data, y canales (D) que dan señales a la data, cual carga instrucciones y son acoplados con los canales B.

BRI proveé dos canales B de 64 Kbps y un canal de señal de 16 Kbps. PRI proveé 30 canales de 64 Kbps y uno de señal de 64 Kbps.

Cable Modem (Telecable)

Los cable modems son una tecnología emergente para proveer acceso de alta velocidad al Internet con el uso de cables coaxial. Un sistema de televisión por cable por lo general tiene un 70 o más canales. Cada canal consume 27 Mbps de capacidad de descarga o downstream y 10 Mbps de capacidad de subir data o upstream. Todos los usuarios deben compartir el total del ancho de banda. El cable modem convierte señales entrantes a formato digital y entonces los conduce al computador a través del adaptador de Ethernet.

Cable Modems Simétricos y Asimétricos

Hay dos maneras de usar un cable modem: simétrico y asimétrico. El modo simétrico permite que las velocidades de subida y de bajar datos sean iguales, este modo es usado muy rara vez. El modo asimétrico es mucho más común. En este caso, el canal de descarga es de más ancho de banda que el canal de subir. Las mayoría de las aplicaciones de Internet son asimétricas por naturaleza. Archivos de imágenes, sonido o video streaming tiende a ser de alto consumo en ancho de banda en la parte de descarga. Al contrario, las peticiones de URL y los mensajes de correo no son de alto consumo en ancho de banda en lo que respecta la subida de datos.

Velocidades de Transferencia de Data

El acceso a cable teóricamente permite velocidades de descarga de hasta 2 Mbps. Su promedio es de 450 KBps, mientras que las velocidades de subida es en promedio alrededor de 300 KBps, la cual refleja los principios fundamentales de esta tecnología. El cable modem no es de tecnología de punto-a-punto como son los modems estándar. Este es del tipo de tecnología bus, significando que todos los usuarios usen el mismo cable y tendrán que compartir su ancho de banda. A diferencia del acceso de un modem estándar, los cable modems permiten uso concurrente de las líneas alimentadoras. La velocidad depende de cuanta personas están conectadas a la vez de la misma vecindad que usan servicio de la misma compañía de servicio de cable. así pues que la transferencia de data no es de un valor constante.

Aunque la velocidad en promedio, que acabamos de mencionar, están muy alejadas de las máximas teóricamente posibles, ellas continúan aun siendo solo aproximadamente siete veces más rápidas que la velocidad de un modem estándar. Y aún más, redes de cable ya son un echo y existen, y por esto ya es relativamente fácil conectar un nuevo usuario a este servicio.

Línea Digital de Subscriptor/Digital Subscriber Line (DSL)

DSL es la alternativa a la tecnología de cable modem. DSL es una tecnología que trae información a alto ancho de banda al usuario final sobre líneas telefónicas de cobre regular. DSL es solo un miembro de la gran familia de los XDSL que incluye diferente variaciones de DSL, como son ADSL, HDSL, y VDSL. Bajo ciertas condiciones (Ej., que tan lejos se encuentra la conexión de la oficina telefónica que proveé el servicio), un usuario puede ser que tenga una velocidad de descarga de hasta 6 Mbps (donde el máximo en teoría es de 8.448 Mbps). El averaje de la velocidad de transferencia va desde 384 Kbps a 1.544 Mbps de descarga (desde el proveedor del servicio hasta el computador del usuario) y alrededor de 123 Kbps de subida (desde el computador del usuario al computador de ISP). Esto es mucho más mejor que las velocidades de un modem y hasta mejor que un cable modem en algunos de los casos. En averaje, cable modems ejecutan en el mismo rango que el DSL.

Tipo de servicio	**Proveedor-usuario (descarga de datos)**	**Usuario-proveedor (carga de datos)**	**Proveedor-usuario (descarga de datos)**	**Usuario-proveedor (carga de datos)**
ADSL	1.5 Mbps	64 Kbps	6 Mbps	640 Kbps
CDSL	1 Mbps	128 Kbps	1 Mbps	128 Kbps
RADSL	1.544 Mbps	1.544 Mbps	1.544 Mbps	1.544 Mbps
ISDL	128 Kbps	128 Kbps	128 Kbps	128 Kbps
RADSL	1.5 Mbps	64 Kbps	6 Mbps	640 Kbps
SHDSL	No soporta	No soporta	768 Kbps	768 Kbps
SDSL	1 Mbps	1 Mbps	2 Mbps	2 Mbps
VDSL	51 Mbps	2.3 Mbps	51 Mbps	2.3 Mbps

¿Cómo esto Trabaja?

Los modems tradicionales utilizan transmisiones análogas. así que, el monto de data máximo que se puede recibir usando un modem ordinario es 56 Kbps. El DSL, por lo contrario, es una tecnología que asume que data digital no necesita ser convertida a análoga y es transmitida directamente al computador en formato digital y permite a las compañías telefónicas usar un ancho superior sobre el mismo alambre de cobre. Parte del ancho de banda puede ser compartido y usado para transmitir señales análogas también. así que, podemos usar la misma línea para conectar tanto el teléfono como el computador para conectarnos.

Tipos de DSL

Como mencionamos anteriormente, existen variaciones del modelo DSL. Aquí mencionamos algunos de los tipos más comunes.

ADSL

La primera variación es la Línea de Subscriptor Asimétrica Digital (Asymmetric Digital Subscriber Line, ADSL), es una forma orientada de uso del hogar y los pequeños negocios, ADSL es llamado asimétrico porque su ancho de banda no es distribuido entre la subida y la descarga de la data. Solamente una pequeña porción del ancho de banda esta disponible en para la subida o los mensaje de la interacción del usuario. La mayoría del ancho de banda esta designado para la descarga de data como la mayoría de la Internet, y especialmente imágenes y multimedia, necesita mucho ancho de banda de baja. Recordemos que la mayoría de usuarios son particularmente sensibles acerca de su velocidades de baja.

CDSL

El Consumidor DSL (Consumer DSL, CDSL) es otra versión comercial del DSL. Es más lenta que el ADSL, pero el divisor (splitter) no tiene que ser instalado del lado del usuario. Esta tecnología es diseñada por la Corporación Rockwell y también esta dirigida a los usuarios del hogar de PC.

HDSL

Una de las primeras variaciones del DSL es el High bit-rate DSL (HDSL), la cual es usada para para transmisiones al nivel corporativo entre la compañía telefónica u el cliente. Esta es una tecnología completamente simétrica, lo cual significa que un monto igual de ancho de banda puede ser usado en ambas direcciones. Por esto es que la máxima descarga de data es menor que la de ADSL.

IDSL

ISDN DSL (IDSL) se encuentra más cercano al ISDN comparado por su velocidad de transferencia que el real DSL. IDSL nos solamente cerca de 128 Kbps de velocidad de bajar.

VDSL

VDSL es DSL de Muy alta velocidad (Very high bit-rate DSL) aún bajo desarrollo y ofrece velocidades de transferencia de data entre distancias relativamente cortas; por ejemplo, puede adquirir hasta 50 Mbps sobre las líneas de 1,000 pies de longitud. VDSL promete las velocidades más alta de todas las variantes del DSL.

Factores que Determinan la Velocidad de Transferencia de Data

El rango máximo de DSL sin repetidores (sin regenerar señal) es de 5.5 km, o 18,000 pies. A medida que la distancia entre el teléfono del usuario y el de la compañía telefónica se incrementa, la velocidad de transferencia de la data disminuye. Otro factor que influye la velocidad de transferencia es el tipo de cobre del alambre. Un ejemplo es, un cable de 24-gauge lleva la misma velocidad de data a una distancia mayor que un cable de alambre de gauge 26. Uno puede mantener la alta velocidad aunque se encuentren a más de 5.5 km.

Por lo general, el DSL es más confiable como el teléfono del usuario. A diferencia del cable modem, la velocidad del DSL permanece consistente a medida que más y más usuarios ingresan al sistema.

X.25

El X.25 fué desarrollado desde el protocolo 1822 del ARPANET, cual fué el esquema de intercambio de paquetes original del ARPANET. El X.25 se convirtió en estándar de ITU en el 1976. Es en la actualidad usado en transacciones por los cajeros automáticos, verificación de tarjetas de crédito, y muchas transacciones de otros puntos de ventas. El X.25 opera una velocidad de 56 Kbps o más lenta. Tecnologías más modernas de intercambio de paquetes (packet-switching) pueden hoy en día ejecutar mejor que el X.25, pero es aún muy usado por todo el mundo. El X25 asegura el transporte de data libre de errores a través de la revisión de errores en muchos puntos de la ruta tomada por la data.

Rápido Intercambio de Paquetes/Fast Packet Switching

Fast packet switching se refiere a dos tipos diferentes de trasmisión a través de redes del tipo maya de intercambio. Con fast packet switching, las tareas de corrección de error, secuencia de paquetes, y reconocimiento no son ejecutados por la red. Fast packet switching es la responsabilidad del sistema destino; y ya que la red tiene menos tareas que llevar acabo puede agilizar la transferencia de la data.

La eliminación de corrección de errores en las capas inferiores ampliamente aumenta los niveles de ejecución. La tecnología de fast packet switching es implementada al nivel de la subcapa de MAC en la capa de Enlace de Data, contrario al X.25, el cual es implementado en la capa de Red.

Ejemplos de la tecnología fast packet-switching are Frame Relay y ATM.

Frame Relay

El Frame Relay es de tecnología fast packet-switching que usa fibra óptica para el cableado digital. Este usa paquetes de longitud variadas (variable-length packets) y permite conexiones a más alta velocidad usando facilidades compartidas de la red. No ofrece soporte extenso de revisión de error y reconocimiento, como si hace el X.25. Como lo implica el nombre, Frame Relay es un servicio de relevo (relay), y el X.25 es un servicio de paquete.

Si su compañía desea conectarse a una red de Frame Relay, su compañía de servicios telefónicos debe conectarlo a un puerto Frame Relay, o Punto de Presencia (Point of Presence, POP). El POP debe ser un proveedor de servicios de Frame Relay. Un puerto de Frame Relay es creado para usted, así dandole acceso a la red Frame Relay.

Redes Frame Relay usan Circuitos Permanentes Virtuales (Permanent Virtual Circuits, PVCs). Estas conexiones lógicas, dedicados, de punto a punto son usadas para la transferencia de data. Una vez establecida una conexión PVC, este existe hasta que la transacción, transmisión o el servicio es terminado. así que, Frame Relay es una red basada en software ya que este comparte la red física con otras redes Frame Relay.

Frame Relay utiliza el ancho de banda en demanda, esto significa que los clientes de Frame Relay pueden elegir el monto de ancho de banda que ellos necesitan. Cada puerto Frame Relay tiene su propia velocidad de puerto, por lo general esta velocidad cae en el rango de 64 Kbps a 1.544 Mbps (T1). Frame Relay es implementado en la subcapa de la MAC de la capa de Enlace de Data.

Modo de Transferencia asincrónico/Asynchronous Transfer Mode (ATM)

ATM utiliza celdas de tamaño fijo (en vez de como las de tamaño variado de paquetes del Frame Relay) y PVC para soportar data así como video y voz. ATM puede ser usado por ambas redes tanto LAN y WAN, pero es más comúnmente usado como un Internet backbone. El protocolo ATM esta definido por el ITU (International Telecommunication Union) y el Foro especializado de ATM.

ATM es referido a una tecnología de relevo de celda ya que esta organiza data en celdas de longitud fija de 53-byte. Estas celdas son transmitida y segmentadas dependiendo del tipo de data que esta siendo enviada. Por ejemplo, el ancho de banda es asignado dependiendo de la aplicación en uso.

Como el ATM usa celdas de tamaño fijo, es más rápido que el Frame Relay. Dispositivos de switching no necesitan ubicar el principio o el fin de cada celda; estas celdas son todas de un mismo tamaño.

Aunque ATM Típicamente opera a velocidades que van desde los 155 Mbps hasta 622 Mbps, este tiene un potencial de velocidad de 1.2 Gbps. Las tecnologías FDDI y T3 están implementando elementos de ATM.

Pero, no todos los medios soportan las capacidades de ATM. Similar al Frame Relay, el ATM es implementado en la subcapa de la MAC de la capa de Enlace de Data.

Sistema T-Carrier

El sistema T-carrier es un formato de transmisión digital Norte Americano que proveé servicios dedicados y de líneas privadas para transmisiones de voz digital y de data a velocidades que llegan hasta los 45 Mbps. Los servicios T-carrier por lo general son usados para conectar una LAN a una WAN, como es la red de una compañía al Internet o a una red Frame Relay.

T-Carrier	Data Transfer Rate
T1	1.544 Mbps
T2	6.312 Mbps
T3	45.736 Mbps
T4	274.176 Mbps

T1

La T1 es un Servicio de alquiler de línea digital COMÚN que proveé un ancho de banda de 1.544 Mbps. Cada línea T1 soporta 24 canales de 64 Kbps. Cada uno de los 24 canales en un circuito T1 puede cargar la transmisión de voz o data.

T1 Fraccional, mejor conocidas como FT1, permite que un cliente alquile canales independiente en vex de una línea t1 completa. Este servicio rinde velocidades de transferencia de data de 1.536 Mbps (un adicional de 64 Kbps es usado para la sobrecarga o overhead).

E-Carrier	Data Transfer Rate
E1	2.048 Mbps
E2	8.448 Mbps
E3	34.368 Mbps
E4	139.264 Mbps
E4	565.148 Mbps

Para conectar la línea T1 a su red LAN, usted necesita los siguientes sistemas:

CSU
Primer punto de contacto del cableado de la T1; este diagnostica y prepara la señal en la línea para la LAN.

DSU
Se conecta al CSU y convierte las señales de la LAN a formato de señales de la T1.

Multiplexer
Proveé un mecanismo para cargar multiple canales de voz y data en una sola línea.

Router
Proveé la interfase entre la LAN la línea T1

T2

T2 es una especificación interna la cual es el equivalente de cuatro líneas T1. Líneas T2 proveen un ancho de banda de 6.3 Mbps. Línes T2 no son ofertadas al publico en general.

T3

Una T3 es equivalente a 28 circuitos T1 y proveen un ancho de banda total de de 44.736 Mbps. T3s fraccionadas permiten al cliente alquilar menos que la total velocidad de una T3 completa. La clasificación de Señal Digital (Digital Signal, DS) usada en Norte America, Japón, Korea, y otros paises proveen una estandarización para los niveles de señales digitales y son sinónimos con el sistema T-carrier. En la siguiente tabla mostramos las velocidades de transferencias del sistema T-carrier.

Sistema E-Carrier

El sistema E-carrier es un formato de transmisión digital Europeo similar al sistema T-carrier Norteamericano. Cada velocidad de transmisión es un múltiplo del formato E1, el cual opera a 2.048 Mbps. En esta siguiente tabla le mostramos el lista de las velocidades de este carrier.

DIRECCIONES DE INTERNET

Como mencionamos anteriormente, para poder pasar mensajes de un computador a otro, cada computador deberá tener su propia dirección de red. Ya hemos visto la dirección MAC, la cual es usada por la capa de Enlace de Data. IP tiene su propio mecanismo de asignación de direcciones. Debido a la separa-

ción de las capas de los protocolos, la capa de Internet no puede por lo general usar ni depender de la capa de Enlace de Data para que le proveé de sus direcciones. Además, el esquema de asignación de IP divide las direcciones en un número de red y un número de nodo, mientras que la dirección MAC no puede.

Cubriremos el formato de las direcciones del Protocolo de Internet, como ellas están estructuradas, y como las direcciones IP son elegidas y obtenidas. También discutiremos las intranets. Los siguientes tópicos son discutidos en esta sección:

- Capa de Internet
- Header (Cabezal) de IP
- Direcciones IP
- Clases de Direcciones de Internet
- Reglas de Direcciones IP
- Eligiendo las Direcciones IP
- Intranets
- Subredes

Capa de Internet

La capa de Internet proveé el medio para la comunicación entre los computadores en una red. Para habilitar esto, la capa de Internet ejecuta la función de enrutar paquetes de una red a otra hasta que estos lleguen a su destino. Para poder efectuar sus funciones de enrutar, la capa de Internet puede asignar direcciones a los hosts en una red y a las redes mismas.

La capa de Internet proveé servicios a la capa más alta de Transporte. Los servicios que proveé están se conciernen por lo general con la transferencia de paquetes entre los hosts.

La capa de Red del modelo OSI cae por debajo de la capa de Enlace y encima de la capa de Transporte. La capa de Enlace de Data puede ser vista como si tuviese dos capas propias. Una de estas, la MAC (Media Access Control), maneja las direcciones MAC de 6-byte que discutimos anteriormente, contiene los manejadores o drivers para las tarjetas adaptadoras de red, y maneja la lógica de colisiones. La otra capa, la capa Control Lógico de Enlace (Logical Link Control, LLC), yace más arriba de ella, establece y termina los enlaces, y establece las direcciones usadas por las varias capas del modelo OSI para comunicarse entre ellas.

La capa de Red proveé servicios a la capa de Transporte (esta comunica sus datagramas), y esta usa los servicios de la capa de Enlace de Data (al transferir paquetes en la red local).

IP Header

La versión actual del IP, la versión cuatro (IPv4), tiene un header que consiste a menudo campos fijos en el header, dos direcciones, y opciones. La longitud del header de un paquete IPv4 es generalmente de 160 bits (20 bytes) al menos que existan opciones. La última versión del IP, la versión seis (IPv6), la discutiremos brevemente.

Aquí le presentamos una ilustración del header del paquete IPv4. El header consiste de seis WORDS (termino informático de almacenaje parecido al CHAR) de 32-bit, la cual es 24 bytes (con opciones presente). Si la data no llena un WORD de 32-bit, particionamiento de bit es a menudo usado para completarlo.

El header del paquete IP contiene varios campos muy importantes:

Versión (4 bits)

Este campo identifica la versión IP, actualmente es la versión 4.

Header Length (4 bits)

Este específica la longitud del header del paquete IP. Valores de la longitud del header son expresados en el número de WORDS de 32-bit en el header, el cual es por lo general cinco al menos que existen opciones presentes.

Service (8 bits)
Este campo indica la confiabilidad, retraso de presidencia, y el resto de los parámetros. También es conocido como el campo del Tipo de Servicio (Type of Service, TOS).

Datagram Length (16 bits)
Este campo define la longitud total del paquete, incluyendo el header, en bytes. La longitud del datagrama no incluye el header usado en la capa de Acceso a la Red (Ej., header del Ethernet).

Datagram ID Number (16 bits)
Esta identificación única de un paquete para los propósitos de fragmentación y reensamblaje. Este número único es copiado en cada fragmento de un datagrama en particular para que así este pueda ser ensamblado.

Flags (3 bits)
Este campo es sólo usado para fragmentación y reemsamblaje.

Fragment Offset (13 bits)
Este indica donde en el paquete este fragmento cae.

Time To Live (8 bits)
Este es medido en intervalos de un segundo, con un máximo de 255. Este campo es conocido como el campo TTL. Los routers por lo normal eliminan un segundo del campo TTL, en vez de calcular los segundos reales, así que el campo TTL es a veces conocido como el campo de salto o HOP Field.

Protocol (8 bits)
Este campo define el próximo nivel de protocolo que es el que recibirá el campo de data en el punto de destino. Si el campo de protocolo es marcado 1, este es un paquete ICMP ; si es 6, este es TCP; si es 17, este es UDP. Fijese que aunque el ICMP y el IGMP son incorporados en la capa de Internet, ellos son encapsulados en paquetes de IP.

Header Checksum (16 bits)
Este campo es usado para error de detección. El checksum calcula solamente el header del IP.

Source Address (32 bits)
Este campo calcula la dirección IP del sistema de origen.

Destination Address (32 bits)
Esta identifica la dirección IP del sistema de destino o final.

Opciones
Este campo indica información del paquete, como son:
- Seguridad/Security
- Enrutamiento de fuente Estricto o Ligero/Loose o strict source routing
- Reportaje de error/Error reporting
- Estampar/Timestamping
- Persecución de errores/Debugging

Por ejemplo, la opción de enrutamiento le permite al que envía especificar la ruta que un paquete a de seguir a través del Internet. Ambas la estricta y la permisivas especifican la vía de enrutamiento.

Enrutamiento permisivo de origen permite que múltiples saltos de red entre direcciones de Internet sucesivos en la lista. El enrutamiento estricto implica que la dirección de Internet especifique la ruta exacta que un paquete debe seguir para llegar a su host de destino; resulta un error si un route no puede reenviar o forward el paquete al nodo especificado.

Direcciones IP

La clave de las capacidades de internetworking del protocolo de Internet esta en su estructura de su direcciones:

- Cada nodo en un internetwork de TCP/IP debe tener una dirección única. Como esta dirección es interpretada por IP, es conocida como la dirección IP.
- Las direcciones incluyen un componente que identifica la red actual a la que el host está conectado además de una porción que identifica el host individual.
- Este concepto puede ser comparado con el de la dirección de una casa, donde parte del componente de la dirección identifica el nombre de la calle (la red) y la otra parte el número de la casa (el host).

Para asegurar que cada usuario en el Internet tenga una dirección IP única, una autoridad central llamada

el ICANN (Internet Corporation of Assigned Names and Numbers), anteriormente conocida como IANA (Internet Assigned Numbers Authority), asigna todos las direcciones de Internet. IANA era financiada y supervisada por el gobierno de los Estados Unidos. La ICANN es una organización sin fines de lucro que opera a nivel internacional, responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de las funciones de gestión o administración del sistema de nombres de dominio de primer nivel genéricos (gTLD) y de códigos de países (ccTLD), así como de la administración del sistema de servidores raíz. Aunque en un principio estos servicios los desempeñaba Internet Assigned Numbers Authority (IANA) y otras entidades bajo contrato con el gobierno de EE.UU., actualmente son responsabilidad de ICANN.

Como asociación privada-pública, ICANN está dedicada a preservar la estabilidad operacional de Internet, promover la competencia, lograr una amplia representación de las comunidades mundiales de Internet y desarrollar las normativas adecuadas a su misión por medio de procesos “de abajo hacia arriba” basados en el consenso.

La mayoría de las direcciones de Internet contiene una parte que es la red y la otra parte que es el host. La parte que es la red precede la parte que representa el host:

porción red.porción host

Las direcciones de Internet son especificadas por cuatro campos, También llamados octetos, separados por puntos:

campo1.campo2.campo3.campo4 o (xxxx.xxxx.xxxx.xxxx)

Valor del BIT	128	64	32	16	8	4	2	1

Ellos son típicamente escritos en una anotación de punto decimal. Cada campo tiene un valor de 0-255 como demostramos en el siguiente ejemplo de una dirección de Internet:

200.42.200.136

10101001	**10100111**	**11001010**	**00000010**
169.	167.	202.	2

En este ejemplo, la parte que representa la red es 200.42.136, y la porción que es el host es el 136. Para ayudar a distinguir entre la parte de la dirección que es red y la parte que es cliente, las direcciones de Internet están divididas en clases, las cuales describiremos en este mismo capítulo más adelante.

Formato Binario vs. Decimal

Las direcciones son llamadas de 32-bit porque cada campo es realmente un byte, y un byte es igual a 8 bits. Como una dirección IP tiene 4 bytes, el total es de 32 bits.

8 + 8 + 8 + 8 = 32

Para poder determinar el valor en bit de una dirección de Internet, la dirección deberá ser convertida desde su formato decimal a si formato binario. Formato binario es la combinación de ceros y unos que las computadoras usan para procesar información. El equivalente binario es determinado calculando el valor de cada bit dentro de cada byte, de izquierda a derecha.

Si el valor binario de una dirección IP es 01111001, usted puede determinar el valor decimal simplemente sumando el valor correspondiente a los bits con valor de 1. Por ejemplo:

01111001 =0+64=32+16+8+0+0+1=121

Tu puedes hacer esto para cada byte en direcciones de Internet de 32-bit. Por ejemplo:

10000011 1110001000001000 11001000 = 131.226.8.200

Ejercicios 2-3: Convertir direcciones de Internet en valores decimales y binarios.

En este ejercicio, usted deberá convertir direcciones de Internet de binarias a decimales y vice versa. Escriba su respuesta en el espacio en blanco. Las respuestas para este ejercicio están en el Apéndice B.

A.- Convertir de valores de binarios a decimal

1. 01111000000000100000011 11110000

2. 11000011 01010101 10011001 11010

B.- Convertir de valores de decimales a binarios

3. 207.199.32.205

Clase A: Rango de 0.0.0.0 al 127.255.255.255			
Valor Binario Inicial	0	Red (1 Byte)	Host (3 Bytes)
126 Redes			16,777,214 Hosts

Clase B: Rango de 128.0.0.0 al 191.255.255.255				
Valor Binario Inicial	1	0	Red (2 Byte)	Host (2 Bytes)
16,384 Redes				65,534 Hosts

Clase C: Rango de 192.0.0.0 al 223.255.255.255					
Valor Binario Inicial	1	1	0	Red (3 Byte)	Host (1 Bytes)
2,097,152 Redes					254 Hosts

Clase D: Rango de 224.0.0.0 al 239.255.255.255					
Valor Binario Inicial	1	1	1	0	Red Multicast (4 Bytes)

Clase E: Rango de 240.0.0.0 al 247.255.255.255						
Valor Binario Inicial	1	1	1	1	0	Experimental/Reservada para uso en el Futuro

3. 151.2.254.60

Clases de Direcciones de Internet

Sin un sistema de clasificación, las 3,720,314,628 direcciones posibles de Internet pueden no tener estructura. Para proveer una estructura, los diseñadores de los protocolos de Internet definieron cinco clasificaciones de direcciones de Internet, de las cuales tres son conocidas y usadas comúnmente:

- Grandes redes, con muchos cientos de miles de hosts, conocidas como redes de clase A.
- Instalaciones de tamaño medio, conocidas como redes de clase B.

• Redes pequeñas con unos cuantos hosts, conocidas como redes de clase C.

Las clases cuatro y cinco están reservadas para usos especiales:
• Multicast, conocidas como clase D.
• Uso futuro y de prueba, conocidas como clase E.

Las cinco clases de redes usan la forma básica de direcciones de Internet. Las clases pueden ser determinadas por la observación a el primer byte de una dirección de Internet.

Las características de cada clase son detalladas a continuación, seguidas por una explicación de cada una:

Antes de leer acerca de las clases de direcciones, por favor note que ni la porción de red ni la porción de hosts de una dirección IP pueda contener todo ceros y ni todo unos en formato binarios. En valores decimales, 255 usualmente significa broadcast, y un valor 0 identifica la red. Estudiaremos más acerca del rol de las direcciones IP más adelante en este capítulo.

Direcciones de Clase A

Las direcciones de clase A usan los primeros 8 bits para la porción de la red y el sobrante de 24 bits para la porción de hosts. Estas proveen el potencial para 126 redes con 16,777,214 hosts. El primer byte específica el número y clase de la red, este puede clasificar de 1 a 126 (127 esta reservado para direcciones loopback). El primer bit de una dirección de una red clase A es siempre un bit 0.

Aquí le presentamos un ejemplo de una dirección de clase A (el primer byte es la dirección de la red):

121.1.1.32

El bit equivalente es:

01111001 00000001 00000001 00100000

Direcciones de Clase B

Las direcciones de clase B usan 16 bits para la porción de red y 16 bits para la porción de hosts. Ellas proveen el potencial para 16,384 redes y hasta 65,534 hosts por cada una. Los primeros 2 bytes especifican el número de la red y la clase, el primer byte puede variar de 128 a 191. Los primeros 2 bits de una red clase B son siempre 10.

Lo siguiente es un ejemplo de una dirección clase B (los primeros 2 bytes son las direcciones de la red):
168.100.1.32

Esto es equivalente a:
1010100001100100000000001 00100000

Direcciones de Clase C

Las direcciones de clase C usan 24 bits para la parte de la red y 8 bits para la porción de hosts. Ellas proveen el potencial para 2,097,152 redes con hasta 254 hosts por cada una. Los primeros 3 bytes especifican el número y la clase de la red, el primer byte puede variar de 192 a 223. Los primeros 3 bits de una red clase C siempre son 110.

Lo siguiente es un ejemplo de una dirección clase C(los primeros 3 bytes son la dirección de la red):
205.96.224.32

El equivalente en bit es:

11001101 01100000 1110000 000100000

Direcciones de Clase D

Las direcciones de clase D soportan multicasting. Con multicasting, un datagrama es dirigido para un grupo que esta identificado por una dirección de red solamente (sin la existencia de porciones de hosts). El primer byte puede variar de 224 a 239. los primeros 4 bits de una red clase D siempre son 1110.

Lo siguiente es un ejemplo de una dirección clase D (los cuatro bytes son la dirección de la red):

230.5.124.62

El equivalente en bit es:

11100110000001010111110000111110

Direcciones de Clase E

Las direcciones de clase E están reservadas para usos futuro. El primer byte puede variar de 240 a 247. Los primeros 5 bits de una dirección de red clase E siempre son 11110.

Podemos ver que pueden haber relativamente pocas y distintas direcciones de Clase A pero muchas direcciones de clase C. En practica, esto es un reflejo exacto de los tipos de redes que se encuentran en el Internet, muy pocas grandes organizaciones y muchas más de las pequeñas.

Es posible de seguir dividiendo una simple red en un número de pequeñas subredes lógicas con el uso de la porción host de la dirección como una dirección de subred. Este topico es cubierto en más detalles en el transcurso del capítulo.

La escasez de números de direcciones IP disponibles para asignar a usuarios nuevos es en la actualidad un tema de discusión y uno de los problemas mayores con IPv4. IPv6 esta diseñado para solucionar estas inconvenientes.

Reglas del Direccionamiento IP

Las direcciones de Internet tienen que seguir fuertes alineamientos para fusionar apropiadamente. Aunque tenemos cubiertos los rangos de direcciones de las clases A, B y C, no todas las direcciones con estos rangos pueden ser usadas como nodo de direcciones de red. Esta sección cubrimos las excepciones.

Todas las reglas del direccionamiento IP están basadas en las reglas básicas de que ni la porción de red o la de host pueden ser todos binarios, unos o ceros.

Direcciones Broadcast

Las direcciones Broadcast son usadas para enviar mensajes a todos los nodos de la red. Las porciones de direcciones IP de red y/o hosts todas son binarias, las cuales generalmente coinciden con el valor decimal 255. Ellas son usadas solo para direcciones de destinación y no pueden ser usadas para direcciones de origen. Existen los siguientes cuatro tipos:

- **Broadcast limitado: 255.255.255.255**

  Ambas, la porción de red y host consisten de unos binarios. Este tipo es usado para configurar hosts cuando estos se inician/bootean. Por ejemplo, una computadora sin una dirección IP puede broadcast para obtener una dirección (por ej., desde un servidor DHCP o BootP).

- **Broadcast dirigido por la Red; netid.255.255.255**

  Este es usado para broadcast a todos los hosts en una red. Por ejemplo, si la cantidad de direcciones IP de tu red es 192.34.200 y la cantidad de hosts es 12, tu computadora puede broadcast mensajes a todos los hosts de la red usando la dirección de destino 192.34.200.255.

- **Broadcast dirigido por la Subred**

  Si una red esta dividida en muchas subredes, un broadcast puede ser limitado a los hosts dentro de una subred. Las subre-

des serán discutidas más adelante en este mismo capítulo.

- **Broadcast dirigido para todas Subredes**

  Si una red esta dividida en varias subredes, un broadcast puede ser enviado a todos los hosts dentro de todas las subredes en la red. Este tipo de broadcast ya es obsoleto, multicasting (ve direcciones de clase D) es preferido.

Direcciones de Red

Las direcciones de red son usadas por los routers para identificar una red. La porción red consiste en la dirección de la red, pero la porción de los hosts consiste de ceros binarios (netid.0.0.0). Por ejemplo, la dirección de red 192.168.3.0 no puede ser usada como una dirección de host.

Caso especial de Direcciones de Origen

El caso especial de direcciones de origen es usado cuando una computadora no tiene una dirección IP. Este es usado solo durante el proceso de inicialización. En un caso especial de direcciones de origen, las porciones de red y los hosts de las direcciones IP son todas ceros binarios (0.0.0.0). Esta dirección es usada cuando una computadora inicializa y pide una dirección IP (por ej: de un servidor DHCP o BootP). Aunque la computadora broadcast una petición para la asignación de una dirección IP, esta dirección de origen es inicializada 0.0.0.0 hasta que se le asigne una dirección IP.

Direcciones Loopback

La dirección loopback 127, no puede ser usada como una dirección de red. Esta dirección permite a un cliente y servidor en el mismo host comunicarse entre si. La dirección loopback es ideal para examinar y diagnosticar. Por ejemplo, si tu computadora es un servidor Web y tu entras "http://127.0.0.1" en un navegador Web (en este caso el cliente), tu puedes accesar al sitio Web (aunque el servidor esta en el mismo sistema). La dirección loopback también puede ser usada para examinar la funcionalidad del TCP/IP con el utilitario ping.

Para sistemas UNiX y NT, la dirección loopback esta listada en el archivo /etc/hosts y es Típicamente 127.0.0.1 con el nombre "localhost" asignado.

Ejercicio 2-4: Determinar las Clases y Direcciones IP Validas

En este ejercicio, usted deberá determinar la clase de cada dirección IP y además que sea una dirección IP válida para un computador. Si no lo es, explicar porque. Las soluciones para este ejercicio están localizadas en el Apéndice B.

	Dirección IP	Clase	Valida? Si o No	Si no es valida, ¿por qué no?
1.	192.23.111.8			
2.	10.1.1.256			
3 .	148.108.62.95			
4.	127.0.0.1			
5.	245.255.123.49			
6.	100.54.100.90			
7.	162.34.0.0			
8.	127.65.18.191			
9.	1.1.1.1			
10.	208.152.84.255			
11.	225.37.257.34			
12.	255.255.255.255			

Direcciones IP Reservadas

El ICANN tiene reservado un espacio de tres bloques de direcciones IP para redes privadas (como esta definida en RFC 1918):

- 10.0.0.0 por medio de 10.255.255.255
- 172.16.0.0 por medio de 172.31.255.255

• 192.168.0.0 por medio de 192.168.255.255

El ICANN sugiere que compañías usen estos IDs de redes si la compañía encaja dentro de una de las siguientes categorías:

- Sus hosts no requieren acceso a otras empresas o hosts de Internet.
- Las necesidades de Internet de los hosts pueden ser manejadas por gateways (ej: gateways de la capa de Aplicación). Por ejemplo, los hosts solo requieren servicios de Internet limitados, tales como correo electrónico, FTP, grupos de noticias y exploradores Web.

Estas direcciones de redes privadas no tienen un significado global. Por lo tanto, se espera que los routers de Internet rechazen (filtren) información acerca de ellos (el rechazo no puede ser tratado como un error de protocolo de enrutamiento).

Los beneficios de usar direcciones de redes privadas incluyen:

- Conservación de direcciones IP globalmente únicas cuando no se requirie.
- Más flexibilidad en el diseño de soluciones empresariales a causa de grandes espacios de direcciones.
- Prevención de choque de direcciones IP cuando una empresa obtiene conectividad a Internet sin recibir las direcciones desde ICANN.

Las desventajas de usar direcciones de redes privadas incluyen:

- Posible reducción de la flexibilidad para accesar a Internet. Si la compañía eventualmente decide prover conexión a Internet para algunos o todos los hosts, se deberá renumerar parte o toda la compañía.
- Si la compañía absor con otra compañía y todos los hosts usan direcciones de red privadas, probablemente se necesitara combinar varias redes privadas dentro de una. Las direcciones dentro de la red privada combinada pueden no ser únicas y tendrá que renumerar los hosts para acomodar direcciones IP idénticas.

Eligiendo Direcciones IP

Cuando la red de un edificio usa TCP/IP, a cada nodo se le debe asignar una dirección. Si la red nunca esta conectada a una gran internetwork, entonces la red puede usar uno de los rangos reservados para uso interno privado.

Si la red es conectada a una gran internetwork, igual que el Internet, los componentes de una dirección de red local deben ser obtenidos desde una autoridad central administrativa. Esta autoridad puede establecerse iniciando routing para la nueva red. Para el Internet, las direcciones de redes de donde anteriormente obtuvieron directamente, desde el InterNIC en Stanford. Este no es el gran caso; dentro del orden para reducir el tamaño de las tablas de routing, en las sinClases InterDominantes de Enrutamiento (Classless InterDomain Routing (CIDR)) las provisiones tienen agrupadas direcciones de Internet por geografía. Ellas están cerradas para que la obtención de cualquier cosa que no sea un grupo de direcciones clase C sea imposible, CIDR hace efectivamente la noción de clases de direcciones redundantes.

Las direcciones IP son básicamente asignadas por tu ISP o tu conexión up-link. Solo los sitios backbone son atractivos para ser concedidas direcciones IP desde el InterNIC. Cada persona debe tomar su dirección IP desde su ISP. Esto se hace un pequeño problema, mientras cambias tu ISP te va a requerir que tu cambies tu rango de direcciones IP en la mayor parte de las circunstancias. Afortunadamente, DNS usualmente hace esto un punto de discusión, así los sitios son usualmente direccionados por nombre no por número.

Una vez que una dirección de red ha sido asignada, las direcciones de los hosts pueden ser colocados por el administrador de la red o un protocolo automatizado como DHCP.

Intranets

Una intranet es la red interna de cualquier organización. El termino "Intranet" fué inventado para referirse a redes internas que usan los mismos protocolos como el Internet, ej: TCP/IP, Este también implica que los basados en IP, los protocolos de alto nivel son usados también, como es el HTTP. Otros usos de los protocolos TCP/IP hay mucha diferencia entre una LAN y una intranet.

Las intranets son distintas, ellas casi siempre implican accesos restringidos para la información y los hosts. Muchas intranets no son conectadas a cualquier sistema de redes exterior, e históricamente esta es como todas las grandes organizaciones edificadas sobre redes de computadoras, usan cualquier tecnología de red que ellos prefieran.

En el mundo moderno de cualquier manera esto incrementa comúnmente para organizaciones que quieren conectar sus intranets a el Internet en una forma limitada, sea para proveer información de ayuda en la intranet a clientes distantes o para permitir a usuarios de su intranet beneficiarse de servicios en el Internet y para distribuir información entre sitios unidos.

Porque la conexión de cualquier red a otras empresas de seguridad unidas es usualmente hecha por la conexión indirecta del intranet al Internet, por ejemplo, usando un firewall. Aquí, el defiende la intranet del resto del Internet mientras permite ciertos tráficos relacionados a través de ciertos servicios.

Aveces organizaciones conectan directamente sus intranet al Internet y luego confían en controles de acceso de servicios personales para restringir quien puede usar sus servicios de intranet.

Las preguntas de todas estas empresas son: ¿Cuáles direcciones IP usar para la intranet? ¿Necesitamos el uso de direcciones que sean únicas en el Internet (con el costo que puedan contraer)?

Direccionamiento de Intranet

TCP/IP es usado en muchas redes que no están conectadas al Internet. Usualmente, donde estas nunca esperan ser conectadas, y una clase de dirección IP apropiada para la red que es seleccionada al azar. Con la explosión de ventajas en el Internet, muchas organizaciones son consideradas una conexión; pero porque direcciones IP deben ser globalmente únicas, ellos se encuentran con el problema de cambiar completamente el esquema de direccionamiento usado en la red.

Un servidor proxy puede ser usado para transmitir conexiones desde una red interna o externa, representando una apropiada forma de traducción de direcciones. Todo aquello que es requerido para una simple máquina. El único hosting, el servidor proxy, tiene una dirección de Internet reconocida en la interfaz externa.

El enrutamiento puede aun ser afectado donde la clase de direcciones usada internamente pertenezca a otra organización que ya este conectada a Internet. Por ejemplo, por accidentes, la dirección IP perteneciente a Micro-Soft puede estar siendo usada. Los enrutadores en la red interna pueden enrutar todos los paquetes destinados para esta clase de direcciones a máquinas locales mejor que a través de el servidor proxy, o el mismo servidor proxy puede enrutar los paquetes despalda a la red interna. RFC 1918 reconoce la practica del uso de direcciones privadas y la codificación del problema por configuración al lado de ciertas direcciones IP. (red clase A 10, redes clase B 172.16 a 172.31 y redes clase C 192.168.0 a 192.168.255). esto es útil porque estas direcciones nunca pueden ser usadas por redes en el Internet.

Aun hay algunas deficiencias en este esquema. Dos organizaciones o departamentos pueden tener usadas algunas clase de direcciones desde este esquema, luego ellos pueden necesitar absorber sus redes (antes de

sobretomar, tal vez); ellos pueden entonces necesitar configurar los hosts con conflictos.

Ejemplo de Direccionamiento de Intranet

En el siguiente ejemplo de direccionamiento de una intranet, la red de Clase A reservada 10.0.0.0 puede ser usada para una gran intranet corporativa. Esto permite un espacio amplio de direcciones para la red interna. De cualquier manera los usuarios querrán aun conectarse a Internet. Esto se logra colocando un gateway entre la intranet, el Internet y varios servicios proxy en el gateway (por lo general este gateway es multihomed).

En el ejemplo anterior el equipo A se hace una consulta a WWW al equipo G. Luego G hace una consulta al Internet y toma la respuesta (proxies). Luego G sirve la respuesta a A.

El gateway toma consultas ingresadas desde el intranet y transmite entonces al Internet usando su propia dirección IP (193.13.5.3- clase C). recibe la respuesta y la toma localmente, ej; al servicio de proxies. La consulta del intranet entonces puede ser satisfecha desde el cache local y las máquinas del intranet no necesitan estar conectadas directamente al Internet.

En el ejemplo anterior, la máquina gateway esta además ejecutando las funciones de firewall (sin permitir ciertos tipos de conexiones) para proteger la intranet de ataques foráneos desde afuera.

Del mismo modo en el ejemplo anterior, la organización sólo necesita una única dirección IP para ser conectada completamente al Internet. En practica, muchos son usados para resistencia y redundancia (un número de gateways), la organización lo más seguro que es dueña de por lo menos una dirección Class C para ser usada con Internet.

Ejercicio 2-5: Direccionamiento de IP

En este ejercicio, vamos a ver las direcciones IP de la red de la Compañía X. No hay soluciones para este ejercicio.

Red de la Compañía X

El administrador de la red de X ahora necesita decidir en un nuevo esquema de direcciones IP para la redes nuevas de la compañía. Se le pide una propuesta para la compañía.

Las necedades expresadas por los departamentos incluyen:

• Trabajadores en la sede principal, realmente sólo necesitan acceso al sistema de la red interna de la Compañía X más la habilidad de navegar el Internet.

• El equipo de desarrollo necesitan acceso completo al Internet.

• El equipo de ingenieros esta preparado para considerar cualquier propuesta pero están preocupados acerca de costos de equipos y quien va a dar soporte a toda la infraestructura de la red.

Preguntas a considerar:

1.¿Debe X usar una red de clase A, B o C o múltiples redes de una clase en particular?

2.¿Cómo se asignaran números de redes a las redes en su diseño?

3-¿Debe X considerar establecer una intranet para ahorrarse direcciones IP?

4. ¿Qué otras opiniones son arrojas de estas consideraciones?

Sub-Redes

Las Sub-redes son una forma muy útil de organizar hosts grupos lógicos dentro de una red. De este modo, una red puede ser dividida en muchas sub-redes. Muchas compañías tienen una sub-red para cada departamento en su organización. Las sub-redes son además útiles cuando los Estándares de la red limitan la habilidad para la red crecer. Por ejemplo, una red ethernet 10BaseT permite un segmento de una longitud de no más de 100 metros, o 328 ft. Para extender la red, podemos crear muchas sub-redes desde la dirección de red existente y conectar cada nodo de la sub-red a un router. Luego, configurar el router para enviar paquetes entre las sub-redes.

El enrutamiento de sub-red permite que numerosas sub-redes existan dentro de una red. Los bits del host son divididos en dos grupos: sub-red y host, por ejemplo, subnetear pide prestado de los 16 bits más bajo (los bits del host) para una red Clase B y los 8 bits para una red Clase C.

Por consiguiente, una dirección de sub-red consiste de las siguientes tres porciones:

Porción de la red | Porción de la sub-red | Porción del host

La única forma de identificar la red, sub-red y la porción de host de una dirección IP es introduciendo un segundo elemento llamado “mascara de sub-red”. La mascara de sub-red es un elemento obligatorio del TCP/IP. Siempre esta configurada con una dirección IP; estas trabajan en pareja en el sistema. La dirección IP en sistema y la mascara de sub-red es el requerimiento mínimo para una configuración TCP/IP.

Mascaras de Sub-redes

Una mascara de sub-red llamada además “netmask” (“mascara” como la llamaremos de ahora en adelante), es un número de 32-bit (similar a una dirección IP) con una correspondencia uno-a-uno entre cada uno de los 32 bits en la dirección de Internet.

Las mascaras de Sub-red sirven dos propósitos principales:

- Para distinguir la porción de la red y el host de una dirección IP.
- Para especificar si una dirección de destino es local o remota.

Distinguiendo las Porciones de Red y Host de una Dirección IP

Las mascaras de sub-red distinguen las porciones de red y host de una dirección IP. Porque el sistema no conoce Cuáles son los bits en el campo de los host a ser interpretados como una parte de la subred de la dirección de Internet, el sistema consulta la mascara de sub-red. La mascara de sub-red le dice al sistema Cuáles bits de la dirección de Internet deben ser interpretados como la red, sub-red y dirección de host.

El tipo más simple de mascara de sub-red es la mascara sub-red por defecto. Por defecto, cada campo de 8-bit es encendido (255- todos los unos binarios) o apagado (0- todos los ceros binarios), dependiendo de la clase de dirección (A, B o C).

La siguiente lista identifica las mascaras de sub-red por defecto para las direcciones de clase A, B y C. Las clases de direcciones D y E no tienen hosts y por tanto no requieren mascaras de sub-red.

- 255.0.0.0 Clase A (defecto)
- 255.255.0.0 Clase B (defecto)
- 255.255.255.0 Class C (defecto)

Especificando si una Dirección de Destino es Local o Remota

Las mascaras de subred especifican si la dirección de destino es local o remota. Note que la mascara de sub-red es usada para enmascarar la dirección de red así solo la dirección del host permanece. En enrutamiento, esto es extremadamente importante. Le permite a un computador determinar si el destino de una dirección esta en una misma computadora (local) o una red diferente (remota).

Si la dirección de destino esta en la misma red, la información puede ser transmitida localmente. Si la dirección de destino esta en una red diferente, la información tiene que ser enviada a un router, el cual puede localizar la red remota.

Ejercicio 2-6: Determinar las Mascaras de Sub-red por Defecto

En este ejercicio, debes determinar la mascara de sub-red por defecto de cada dirección IP. Las soluciones para este ejercicio están localizadas en el Apéndice B.

1. 17.223.13.222
2. 194.10.99.2
3. 211.34.126.10
4. 152.4.202.69
5. 128.156.88.1

Interfases de Red

Hay varios tipos diferentes de interfases de red utilizadas para conectarse a una red. Algunos tipos comunes de interfases incluyen la interfaz loopback, Ethernet, Token Ring, SLIP, PPP y plip. Puede haber más de una interfaz para cada tipo listado, en cualquier caso, las interfases son enumeradas consecutivamente iniciando con 0. Por ejemplo, eth0 designa la primera interfaz Ethernet.

Los siguientes tópicos son discutidos en esta sección:
- Nombres de Interfases
- ifconfig
- Alias de Interfases
- Configuración Dial-Up
- Configuración de la Red
- Scripts de Inicio de Red
- Herramientas de Configuración
- Probando Conectividad con ping

Nombres de Interfases

En esta sección, cubriremos algunas de las interfases de redes más comunes.

lo

La interfaz lo de red es conocida como la interfaz loopback. El propósito de lo es direccionar el tráfico que se origina desde la máquina local que además tiene la máquina local como destino final. La dirección loopback tiene una dirección IP especial que esta asignada para tráfico local. Esta dirección es 127.0.0.1. Todo computador ejecutando GNU/Linux necesita tener una interfaz lo configurada y activa.

eth

La interfase eth es usada para conectarse a una red ethernet. Las redes ethernet son los tipos más comunes de interfases usadas para conectarse a una LAN debido a que ellas alto nivel de transmisión de información, permiten múltiples equipos, y son fáciles de implementar. Si tienes una tarjeta de red, lo más probable que esta sea un tarjeta ethernet.

tr

Una interfaz tr designa una interfaz para un Token Ring. Las redes Token Ring son muy útiles ya que ellas evitan el problema de colisiones de data de las Ethernet. Este tipo de LAN no es tan común como el Ethernet ya que las redes Token Ring no son tan fáciles para implementar.

SLIP

Las interfases SLIP son interfases de líneas seriales. Las interfases seriales son comúnmente usadas para conectar computadoras aisladas a otras redes de computadoras mediante un modem, pero ellas pueden además ser usadas para conectar dos máquinas directamente. Un problema que existe cuando se usan líneas seriales para conectar a una red es que la transmisión serial de información no es muy rápida debido a limitaciones de equipo (hardware).

PPP

Otra interfase de línea serial es la ppp a excepción que usa el protocolo de punto a punto (ppp). Una interfaz PPP es creada cuando un enlace (link) serial es establecido usando PPP. Este tipo de interfaz puede ser usado para los mismos propósitos como la interfaz SLIP, pero una interfaz PPP es mucho más confiable que una interfaz SLIP y su uso es más popular.

plip

Las interfases plip son similares a las interfases SLIP a excepción de que ellas son ejecutadas sobre líneas paralelas. El uso típico de una interfaz plip es para conectar dos máquinas por medio de puertos paralelos. Una interfaz plip puede transferir información más rápido que una interfaz de línea serial, pero los periféricos adheridos a el puerto paralelo no pueden ser usados mientras la interfaz plip este activa. Por ejemplo, activando la primera interfaz plip desactivaría lp0 en favor de plip0. Debe notarse que una interfaz plip no puede ser inicializada con un cable paralelo ordinario, se necesita un cable especial llamado Null Printer.

ifconfig

El comando ifconfig es usado para configurar las interfases de red en el sistema, ifconfig hace dos funciones básicas. Este puede listar interfases y puede ser usado para configurar más interfases. Las interfases que están activas comúnmente en el sistema pueden ser vistas escribiendo el siguiente comando:

ifconfig

El estado de una interfaz en específico puede ser observado al invocar ifconfig con el nombre de la interfaz como el único argumento.

ifconfig eth3

Para hacer algo más que desplegar las interfases con ifconfig, el usuario debe tener privilegios de root. El uso más común de ifconfig es para activar y desactivar una interfaz. Esto se hace escribiendo “ifconfig” seguido por el nombre de la interfaz tal como eth2, luego “up” o “down”, up específica que interfaz debería ser activada, mientras que down indica que la interfaz debería ser desactivada. Por ejemplo, el siguiente comando:

ifconfig eth2 up

Este comando activa la segunda interfaz ethernet.

Una dirección IP puede ser asignada a una interfaz usando el comando ifconfig. Cuando se usa de esta forma la estructura del comando ifconfig es:

#ifconfig [dispositvo] [direccionIP] netmask [número netmask] [up\down]

Un ejemplo de configuración de una dirección IP y activar una interfaz es:

ifconfig eth0 192.168.2.7 netmask 255.255.255.0 up

En este ejemplo, la interfaz eth0 es configurada con la dirección IP 192.168.2.7 y con una mascara de 255.255.255.0. El uso de up en el comando activa la interfaz. Esta opción puede ser omitida porque es la acción por defecto. Para desactivar la interfaz de red se escribe:

ifconfig eth0 down

Mientras se configuran las interfases de red, el kernel asume ciertos valores por defecto. En el ejemplo anterior, han sido especificadas la dirección de red y la mascara. Si alguna de estas direcciones no hubiese sido proveída, el kernel tomaría unos valores por defecto basados en la información que si fué suplida. Si la mascara no fué suplida, el kernel lo asumiría como que es una red Clase C que se está configurando y la configuraría como una dirección de red de 192.168.2.0 y una dirección de broadcast de 192.168.2.255 para la interfaz.

Hay muchas opciones de línea de comando para controlar la funcionalidad de una interfaz. Algunas de las opciones del comando ifconfig son las siguientes:

up	Esta opción activa la interfaz.
down	Esta opción desactiva la interfaz.
[-]arpa	Esta opción activa o desactiva el uso de arpa en la interfaz.
[-] promisc	Esta opción activa o desactiva el modo promiscuo de la interfaz. Si esta especificada, todos los paque tes en la red pueden ser recibidos por la interfaz.
[-]allmulti	Esta opción activa o desactiva la recepción de todos los hardware multicast packets.
mtu N	Esta opción le permite configurar la máxima Unidad de Transacción (Maximum Transaction Unit (MTU)) de este dispositivo.
netmask <addr>	Esta opción permite al usuario configurar la mascara de red perteneciente al dispositivo.
[-] broadcast [addr]	Esta opción nos permite activar, desactivar y configurar la aceptación de datagramas para la dirección broadcast específica.
[-] pointopoint [addr]	Esta opción nos permite configurar la dirección de las otras máquinas al final de la longitud del punto a punto, tales como slip o ppp.

Además están disponibles opciones avanzadas para controlar el comportamiento de la interfaz, tales como las configuraciones de hardware, asignación de protocolos a la interfaz y opciones para configurar una interfaz para usar IPv6.

Alias a Interfases

Es posible asignar múltiples direcciones IP a una misma interfaz. Este proceso es conocido como crear alias de IP. Un beneficio de tener una dirección IP asignada un alias es que le permite múltiples direcciones IP apunten a una misma localidad Física. Esto puede ser muy útil para establecer un host en dos redes separadas sin la necesidad de dos dispositivos de red. Uno puede además querer asignar a un único host múltiples direcciones IP y así poder ejecutar múltiples servidores pero con direcciones diferentes, está técnica es que hoy día hace posible tales como los servidores Web.

Tradicionalmente, un host multihomed es uno con múltiples interfases. Cada interfaz debe tener una dirección IP distinta. Esta interfaz puede ser de cualquier forma o conexión tal como una tarjeta Ethernet, DSL, un vinculo punto-a-punto o un cable modem. De este modo un host que tenga múltiples interfases sería una máquina con un vínculo Ethernet y uno punto-a-punto. Un enrutador es un dispositivo multihomed porque envía datagramas que llegan en una interfaz y entregan en una interfaz diferente. Un host multihomed no es un router a menos que remita los paquetes. Cuando se cuenta el número de interfases para ver si un host es multihomed, la interfaz loopback no es incluida. Esta forma de multihoming ha pasado a ser obsoleta y una nueva generación de multihoming se ha popularizado.

Una interfaz alias es similar a una interfaz regular excepto que otro número es adherido al nombre de la interfaz para designar el número alias. La interfaz eth0:0 es la designada primer alias de eth0. El interfaz eth0 es la misma interfaz física que eth0:0 excepto con una dirección IP diferente. Para que los alias funcionen

después de reiniciar, los pasos de la configuración deben colocarse en un shell script que se ejecutará cuando se inicie el sistema.

Para configurar un IP alias, el kernel debe tener soporte de alias de IP, ya sea dentro del kernel o como un módulo. El siguiente comando puede ser usado para cargar el módulo de alias de IP:

/sbin/modprobe ip_alias

La interfaz que se le va a aplicar el alias debe existir en el sistema. Por ejemplo, antes de que se pueda crear eth0:0, la interfaz eth0 debe existir. Esto se puede hacer mediante las herramientas de configuración de las distribuciones específicas o se puede hacer manualmente. Los permisos de Root deben estar disponibles para permitir estas acciones.

Primero, la interfaz debe ser asignada su propia dirección IP con ifconfig. Una vez la interfaz obtenga el número de IP, esta empezará a interactuar con la red. Cuando la interfaz este lista y funcionando se debe usar ifconfig nuevamente para asignar los números IP alias. Con un comando similar al siguiente:

ifconfig devx:y dirección-ip up

creará el alias y para el dispositivo m, asignando la apropiada dirección IP, entonces funciona la interfaz. Por ejemplo:

ifconfig eth0:0 152.168.1.20 up

Iniciaría activado un alias de eth0 con el la dirección IP de 192.168.1.20.

Aunque este ejemplo solo muestra la adicción de un alias, cada interfaz de red puede por defecto aguantar un máximo de 256 alias. El número por defecto esta definido en /usr/include/linux/net_alias.h.

Para borrar un alias, se usa la siguiente línea:

ifconfig eth0:0- 0

El espacio después del eth0:0 indica eliminar el alias y esto se puede hacer para cualquier alias. Para guardar los cambios (alias creados y/o borrados) al reiniciar, hay que agregar las líneas para iniciar o matar los alias a un script que se ejecute cuando el sistema se inicia, tal como el archivo /etc/rc.d/rc.local.

Configuración Dial-Up

Para sistemas que no estén conectados a una red, muchas veces la única forma para conectar a otras máquinas es mediante una conexión serial, como es un modem. El protocolo usado cuando la red esta mediante líneas seriales es el Protocolo Punto-a-Punto (Point-to-Point Protocol (PPP)). Cuando se inicia una conexión serial, pppd es usado para establecer la conexión. Para conexiones seriales sobre líneas ISDN, se usa una versión especial de pppd, ipppd.

Demonio del Protocolo Punto-a-Punto (Daemon pppd)

Mientras TCP/IP y otros protocolos son más comunes en el Internet, PPP es aún el protocolo usado para la conexión dial-up. GNU/Linux tiene un demonio que interactua con el resto de el Internet mediante PPP. El pppd tomará los paquetes desde una red y lo convertirá entonces en un formato aplicable para transmisiones PPP. Este demonio esta ubicado en /usr/sbin/pppd. Hay muchas formas para iniciar un demonio. Puede ser iniciado manualmente desde la línea de comandos con scripts o usando un programa por separado para hacer las conexiones, como era el dip. El dip (Dial-Up IP Protocol Driver) puede ser usado como un programa automatizado o un utilitario interactivo para conectar el modem a una conexión e iniciar el pppd. Otras formas es usar los utilitarios gppp y kppp para conectar automáticamente.

Para configurar y subir pppd en una máquina, hay dos partes a configurar. Primero, debe configurar PPP en el kernel. Esto es normalmente una opción por defecto cuando el kernel es inicialmente configurado. Para

revisar, debe observar el mensaje al iniciar y observar las líneas que contienen información acerca del PPP. Si PPP no esta configurado en el kernel deberá hacerlo antes para que el pppd trabaje. Para configurar e instalar el pppd, puede usar un archivo RPM o un tar.gz de fuente. La información importante que debe conocer para la instalación es el tipo de modem, velocidad y cual dispositivo este usa en GNU/Linux.

El pppd debe estar ejecutándose para poder usar el PPP. El PPP es un sistema en tres partes y necesita el pppd para controlar cada aspecto de el PPP. Las tres partes del PPP son un formato estándar de data para enviar información a través líneas seriales, un Protocolo de Control de Vínculos (Link Control Protocol (LCP)) y una colecciones de Protocolos de Control de Red (Network Control Protocols (NCPs)). El formato estándar usado para enviar información es una trama. Las tramas son paquetes de data que contienen un bloque cabezal (el cual identifica la trama) y un block de data (la información actual que necesita ser transmitida). El LCP es usado para transportar las tramas que controlan el PPP y establecen, mantienen y terminan la conexión. NCP es el protocolo de configuración para los protocolos de la capa de Red. Hay un NCP individual para cada protocolo de la capa de Red que es soportado por PPP.

Para usar pppd, debe haber información disponible para el demonio. Algunos de los diferentes tipos de información que el demonio necesita saber incluyen el tipo de modem usado, si usan y que tipo de autentificación esta presente y cual NCP usan (por lo general es el IP Control Protocol (IPCP)). El pppd es bastante versátil, con muchas opciones que pueden ser llamadas al inicio del demonio. Además de las opciones configurables listadas previamente, algunas opciones incluyen la velocidad de conexión, donde se encuentra el script de conexión, si usar una ruta dedicada mediante las tablas de enrutamiento una vez que la conexión este establecida, si se necesita cerrar (lock) el dispositivo en uso por el demonio, que tan grandes pueden ser los paquetes recibidos (Maximum Receive Unit (MRU)]) y que tan grandes pueden ser los paquetes enviados (MTU).

El MRU específica el tamaño de la trama más grandes que pueden ser recibidas en una conexión. El MTU es el tamaño más grande de las tramas que pueden ser enviadas en una conexión. La medida por defecto para ambos casos es 1,500 bytes. Una medida alternativa es de 576. Estas medidas son negociables al inicio de una sesión entre dos puntos finales de la conexión.

El archivo de configuración /etc/ppp/options es usado por todo el sistema para la configuración del pppd. Un ejemplo de un archivo de /etc/ppp/opcions es:

```
$  Log PPP control   framss  to syslog
debug
1 Serial  interfase options
/dev/ttysl  33600
lock
modem
crtscts
$  User name  for PPP authentication
user myaccount
remotename itiyisp
$ Use PPP peer as default gateway in system routing  table
defaultroute
```

Este archivo de configuración muestra un usuario quien esta marcando a través de la interfaz serial /dev/ttys1 a 33600 bits por segundo. El control de hardware (crtscts) y el control de modem (modem) están habilitados. La opción de la ruta por defecto (defaultroute) establece que el otro punto de la conexión PPP es el gateway por defecto para todo tráfico saliente. Alguna de las otras opciones que son posibles incluir en el archivo son: asyncmap, auth, connect, lock, idle y ktune. La opción asyncmap específica Cuáles caracteres necesitan ser escapados para una transmisión correcta. La opción auth requiere que el compañero (peer) se

autentifique antes de que los paquetes puedan ser aceptados o enviados. Connect es una opción que dice Cuáles scripts usar para configurar una línea serial. Lock, cuando se habilita, puede bloquear el dispositivo serial, así que solo el pppd lo puede usar. Idle puede especificar el número de segundos suspendido (idle) permitidos antes de que se efectúe una desconexión automática. La opción ktune es usada para permitir a pppd cambiar el kernel como sea necesario mientras se ejecuta.

Cuando se inicia el pppd, solo los errores fatales se muestran en pantalla. Los errores que no son fatales se almacenan en el archivo /var/log/daemon.

Aunque PPP es usado principalmente sobre conexiones dial-up, es posible usar pppd sobre líneas permanentes. Un demonio separado se ha escrito para usar PPP sobre ISDN, llamado ipppd.

Demonio de Protocolo Punto-a-Punto de ISDN (ipppd)

Para líneas ISDN, hay un demonio separado que es usado que no es pppd. El demonio ipppd es para sistemas GNU/Linux que se conectan sobre ISDN. De cualquier manera el usuario tiene que decidir si usa una conexión PPP sincrónica o asincrónica. Si el PPP en uso es asincrónica, lo mejor es usar el pppd normal con dispositivos ISDN y no el ipppd. Los emuladores existen para probar dispositivos que corren sobre líneas ISDN para transmisiones asincrónica que trabajan con el pppd estándar.

Igual que el pppd, ipppd tiene un archivo de configuración global: /etc/ppp/ioptions. Este archivo puede permitir al administrador controlar las opciones que son aceptadas por ipppd. Muchas de las mismas opciones de pppd son además usadas por ipppd, tal como nombre de dispositivo(s), si usar autentificación y si el dispositivo(s) tiene que ser bloqueado exclusivamente para el uso del ipppd. De cualquier manera, es importante recordar que algunas opciones que so necesarias para el pppd no lo son en una conexión ISDN. Esto incluye velocidad de transmisión, que hacer después de una desconexión y cualquier manejo de transmisiones asincrónica.

El ipppd puede manejar más que un dispositivo que este conectado mediante líneas ISDN. Pero, el ipppd es solo iniciado una vez, sin importar cuantas conexiones se están ejecutando.

El ipppd además configura y transmite LCP y NCP para las transmisiones. El LCP es usado para transportar las tramas de control (paquetes de datos) para la conexión. Los NCPs son los protocolos para las diferentes protocolos de la capa de Red que pueden estar presente en una conexión PPP.

Configuración de Red

Hay más en configurar en una red que activar una interfaz y asignarle una dirección IP. La resolución de nombres y enrutamiento son exactamente dos de muchos componentes necesarios que conforman el funcionamiento y uso de una red.

Enrutamiento

Después de que el dispositivo ha sido asignado su dirección IP y ha sido puesto a funcionar, el comando route debe ser usado para establecer el enrutamiento para la nueva interfaz si se esta usando enrutamiento estático. El comando exacto para configurar una ruta por defecto puede variar dependiendo en la configuración específica de la red, pero la forma general del comando se muestra a continuación.

route add opciones

Las opciones varían dependiendo en la configuración particular de la red a la que el sistema este conectado, pero las partes route y add del comando son requeridas. Algunas opciones que pueden ser necesarias son el gw para establecer el gateway por defecto, -net para establece una ruta a una red o -host para establecer una ruta a otro host. Por ejemplo:

route add default gw 192.168.2.7

significaría que todos los paquetes destinados para el exterior de la red puedan usar 192.168.2.7 como el gateway por defecto.

Resolución de Nombre

Para identificar una interfaz de red, un servidor de nombre (name server) debe establecerse. Este asignará un nombre de dominio a la dirección IP y es establecido editando el archivo /etc/resolv.conf. Un ejemplo de que debe aparecer en el archivo es similar a las siguientes dos líneas:

domain abiertos.org
nameserver 192.168.2.7

La línea domain designa específicamente el nombre del dominio del sistema local. La segunda línea específica la dirección IP del servidor de nombre primario que debe ser usado. Múltiples servidores de nombres pueden ser usados en caso de emergencia que el servidor primario no pueda ser accesado.

El comportamiento de la resolución de nombre puede ser afectado aún más editando el archivo /etc/host.conf. Normalmente este archivo contiene las siguientes dos líneas para permitir un archivo hosts local:

order hosts.bind
multi on

La primera línea es este archivo dice que cuando se busca un nombre, el archivo de los hosts locales es revisado primero y entonces el Sistema de Nombre de Dominio (Domain Name System (DNS)) es revisado de acuerdo con las instrucciones en el archivo /etc/resolv.conf.

El siguiente paso en manualmente configurar la interfaz de red implica editar el archivo /etc/hosts. Este archivo contiene ambos, los nombres y las direcciones IP de todos los hosts locales. Una vez que un host es colocado en este archivo, el servidor de nombre de dominio no tiene que ser consultado para obtener su dirección IP. Pero recuerde que, este archivo debe ser actualizado en caso de cambio de una dirección IP a uno de los hosts. Si el sistema es bien administrado, los únicos nombres de host que deben aparecer en este archivo son interfaz loopback, y el nombre del host local. Una entrada típica en este archivo es:

# <Dirección IP>	**<nombre host>**	**<alias>**
192.168.2.2	**eq2.abiertos.org**	**dos**
192.168.2.7	**eq7.abiertos.org proxy**	**siete**

Todas las líneas anteriores inician con la dirección IP, luego la segunda parte es el nombre del host, seguido por un alias (si alguno) para ese host. Generalmente, los alias no son requeridos, pero la única ventaja de tenerlos es que las máquinas pueden ser accesadas en la red local escribiendo el nombre alias cortos en lugar de el nombre completo del host. En el ejemplo, eq7.abiertos.org puede ser accesado localmente como proxy.

Scripts de Inicio de la Red

Puesto que hay muchos pasos para establecer y configurar una interfaz de red, algunas distribuciones han tomado pasos para facilitar la configuración de interfases de red de todo tipo. Estos scripts son más útiles cuando se usan interfases como las de ppp que necesitan ser actividades y desactivadas con más frecuencia. Estos scripts se encuentran en el directorio /etc/sysconfig/network-scripts.

El directorio /etc/sysconfig/network-scripts contiene tres diferentes tipos de archivos: archivos de configuración, archivos de activación y archivos de desactivación. Los archivos de configuración son de la forma ifcfg-interfase y contienen información como son la dirección IP, la dirección y mascara de red. Los archivos de configuración pueden además contener parámetros para activar una interfaz en el inicio y algunos otros parámetros específicos a la interfaz.

Los archivos de activación siguen la convención de nombre ifup-interfase y contienen código para activar la interfaz específica. Los archivos de desactivación son usados para desactivar una interfaz y son de nombres parecidos a: ifdown-interfase.

Además de los tres tipos de archivos previamente listados, hay otros archivos especiales para asistir las configuraciones de red. El más importante de estos archivos es network-functions. Este archivo no es un shell script y contiene cierta importantes informaciones para los scripts. Otro script especial es ifup-aliases y es usado para configurar alias para interfases en el sistema. Antes que una interfaz pueda ser asignada alias, la interfaz debe existir físicamente, estar configurada y activa. El último shell script es ifup-routes y es usado para configurar enrutamiento estático para el sistema. Para ifup-routes ser útil, la red tiene que estar emplear enrutamiento estático.

Cuando se usan estos dos últimos scripts (ifup-aliases/ifup-routes), es importante notar que ellos solo pueden ser usados en sistemas Red Hat desde que ellos hacen llamadas a programas de administración específico a Red Hat. Ellos además, deberán ser ejecutados como root ya que ellos requieren los privilegios del superusuario para ejecutar estos programas de administración. Ejecutarlos con el comando sudo puede resolver este problema.

Una manera de aprovecharse de los scripts de inicio de red es usando los comandos ifup e ifdown. Estos scripts, ubicados en /sbin, pueden ser usados para activar o desactivar una interfaz. Estos permiten al sistema mantener las configuraciones actualizadas sin la asistencia del administrador. Los cambios pueden ser configurados y almacenados usando utilitarios de administración tal como netcfg y network-admin de gnome. A veces es buena practica en el sistema permitir a ciertos usuarios, que no sean administradores, activar y desactivar algunas interfases, especialmente interfases ppp y SLIP. Por ejemplo, podemos usar el siguiente comando para inicializar una conexión PPP:

$ /sbin/ifup ppp0

Herramientas de Configuración

Hay un número de herramientas disponible para configurar las interfases de red. El único común de todas las distribuciones GNU/Linux y las mayoría de variantes de UNiX es ifconfig, una herramienta de línea de comando. La mayor parte de las grandes distribuciones de GNU/Linux tienen además sus propios programas gráficos o basados en menu de configuración. Los dos desktops más usados también proveen interfases gráficas para la administración de los dispositivos de red de todo tipo. GNOME usa network-admin y KDE tiene a Kcontrol estos dos tools hacen a los archivos y producen configuraciones más fácil luego de mantener. Pero si alguien eligió trabajar en MANDRIVA, SuSE, RedHat o cualquier de las otras grandes distros de GNU/Linux debe acojerse a sus herramientas de configuración ya que esto es de las principales razones que ellas ofrecen para elegirlas como plataforma de trabajo.

Aquí en esta sección discutiremos las diferentes herramientas de configuración de interfases de red proveídas por las principales distros. No es en ninguna manera la intensión de cubrir todos los aspectos o de ser completo a todas distros y a toda herramienta.

YaST2 (SuSE)

Yet Another Setup Tool 2 es una herramienta basada tanto en GUI (Interfases Gráficas de Usuarios) o MDUI (Interfases de Usuarios Manejadas por Menus) manejado a través de Menus que es la parte integral de la distribución SuSE GNU/Linux. YaST2 es la herramienta principal para la instalación, configuración y administración de hosts ejecutando SuSE. Esta sección trataremos sólo con la capacidades de configuración de red YaST2.

Para accesar a YaST2, hay que iniciar en el host como root y seleccionar YaST2 en el menu de KDE o

desde un terminal. Cuando la ventana de YaST2 aparece, el usuario puede seleccionar la administración del sistema desde el menu principal y luego la configuración de la red en la ventana pop-up que aparece. Esto permite hacer la configuración a los host de la red. El usuario se le presentan muchas opciones, sin embargo sólo unas cuantas pertenecen directamente a la configuración de la interfaz de la red.

La configuración base de la red es donde las direcciones IP son configuradas para las interfases de red identificadas en el sistema. Aquí se les puede asignar direcciones IP a los dispositivos de red y pueden ser activados o desactivados. Aquí en este punto, podemos elegir que tipo de red estamos configurando (tal como Ethernet). La opción cambiar el nombre del host es usada para configurar el nombre del host así como el nombre del domino al que el host pertenece.

La opción de cliente DHCP se encontrará activada sólo si el paquete dhclient ha sido previamente instalado. DHCP es usado si el host recibe las direcciones IP y la información de enrutamiento dinámicamente en la red a la que este conectada.

La opción de configurar parámetros ISDN esta disponible solo si el paquete i4l ha sido instalado. Esta opción permite a los usuarios configurar las opciones que su tarjeta ISDN usa para conectarse al proveedor de servicios. Esta opción fija la configuración para elementos tales como tipo de red, host y números de teléfono remotos, número de llamadas recuperadas, tiempo de espera entre las llamadas y login/clave.

Netconfig (Red Hat/Fedora)

Los usuarios de la distribución Red Hat GNU/Linux tienen el programa netconfig para ayudar en la configuración de una red. Netconfig puede ser usado para configurar muchas características de la red, incluyendo ambas tareas cliente y servidor.

El propósito del utilitario netconfig es permitir a usuarios configurar su red TCP/IP desde cero. Netconfig provee un GUI a través del cual casi todos los aspectos de los sistemas de la red pueden ser definidos y controlados. Los usuarios son permitidos controlar casi todas las facetas de la red de su computador desde la información del host y enrutamiento hasta Apache y Samba.

Primero, para que los usuarios puedan tener acceso a netconf, los usuarios deben tener privilegios de root. No es buena práctica permitir a cualquier usuario en un sistema editar o controlar funciones básicas de la red, puesto que podrán interrumpir fácilmente el sistema de la red. Una vez que el usuario ha accesado como root, netconf es muy fácil de iniciar. Este se encuentra en el directorio /usr/sbin y puede ser lanzado simplemente escribiendo en la línea de comandos la siguiente sentencia:

```
# /usr/sbin/netconfig
```

Después de que netconfig ha sido iniciado, la ventana de netconf se presenta en el escritorio. Dentro de la ventana principal se encuentran tres tabuladores: tareas de cliente, tareas de servidor y miscelaneo.

Tareas del Cliente

La sección de tareas de cliente de netconf contiene las herramientas necesarias para configurar las opciones de red del host local. La ventana Información Básica del Host permite al usuario entrar el nombre del host de la computadora y configurar cualquier adaptador que este adherido al sistema.

Especificación de Nombre del Servidor es usada para especificar el nombre del dominio local, nombre de servidores usados por el host local y en Cuáles dominios buscar cuando el nombre de un sistema este pasado como local (un nombre que no incluya un dominio).

Las opciones de Enrutamiento y Gateways toma incontables selecciones, todas de las Cuáles permite a un usuario configurar diferentes rutas y gateways a otras redes o hosts. Esta parte del utilitario además otor-

ga acceso a la configuración del demonio routed, el cual es usado para redes complejas que interconectan grandes redes.

La Ruta de Búsqueda de Nombres de Host (Host Name Search Path) configura como los diferentes servicios de nombres serán probados por el host. Las tres opciones son hosts para el archivo /etc/host, NIS para usar el Network Information System y DNS para usar el Sistema de Nombre de Dominio. Los tres nombres de servicios pueden ser probados en cualquier orden y es posible seleccionar una configuración donde uno o dos de los servicios no sean probados completamente.

La sección de Ajuste de Interfaz IPX permite configurar el host para usarse en una red IPX, mientras el botón PPP/SLIP/PLIP hace lo mismo para las redes PPP/SLIP/PLIP.

Tareas de Servidor

La sección (Server Task) de Tareas del Servidor de netconfig es para configurar el host local para usarse como un Servidor en la red. a través de la asignación de alias de IP para hosts virtuales, es posible asignar múltiples direcciones IP a un dispositivo de red. Esta es la sección donde los alias son configurados.

Misceláneos

Esta última sección principal tiene dos partes importantes: información acerca de otros hosts e información acerca de otras redes. La información acerca de otros hosts y la información acerca de otras utilidades de redes muestran la información en los archivos /etc/hosts y /etc/networks, respectivamente. Ellos permiten modificaciones de entradas existentes así como la adicción de nuevas entradas.

Netconfig (SlackWare)

Contrario a las tres otras herramientas discutidas en esta sección, netconfig es realmente un script que esta diseñado para ayudar al usuario novato en la configuración e instalación de los archivos de configuración de TCP/IP necesarios para conectar el host a una red. Netconfig le hace una serie de preguntas al usuario y luego crea el archivo rc.inet1. Este archivo es usado por el host para configurar los dispositivos de red adheridos al sistema y enrutando. Netconfig es además capáz de probar y activar la tarjeta de red para el sistema.

Netconfig es iniciado escribiendo /sbin/netconfig en la línea de comandos. Este sólo puede ser llevado acabo por el super usuario root porque netconfig hace cambios a archivos de configuración del sistema.

Una vez que netconfig esta ejecutándose, la primera información que deberá suplir es el nombre del host. Este es el nombre de la máquina que netconfig se está preparando para configurar. La entrada no incluye el nombre de dominio del host. El nombre de dominio es pedido próximo y no incluye el (punto) delantero “.”.

Luego, netconfig puede preguntar si el usuario desea utilizar TCP/IP a través del loopback. Esto solo se puede hacer si el usuario va ha usar el host únicamente como una máquina aislada (stand-alone). Si el usuario quiere usar el host en una red, la respuesta a esta pregunta debe ser no.

La siguiente serie de preguntas son acerca de números IP que serán usados por el host para comunicarse en la red apropiadamente. La primera es la dirección IP para la máquina local, la segunda es la dirección para el gateway y la tercera es la mascara de red. Toda esta información se necesita para que el host trabaje apropiadamente en la red.

Finalmente, netconfig le preguntará al usuario que servidor de nombre será usado. Si el usuario responde afirmativamente, entonces netconfig le permite al usuario entrar la dirección de servidor de nombre. Si necesita ingresar un segundo y tercer servidor de nombres, el usuario puede adherirlos editando el archivo /etc/resolv.conf.

Los archivos de configuración /etc/rc.d/rc.inet1 y /etc/rc.d/rc.inet2 pueden ser modificados cuando sea para cambiar la configuración de la red. Como se estableció previamente, el archivo rc.inet1 configura los dispositivos de red y configura el enrutamiento. El archivo rc.inet2 no hace el lanzamiento actual de aplicaciones y demonios que corren el la red, incluyendo el demonio inetd que ejecuta en la mayor parte de los otros servicios de red.

WvDial

WvDial es una aplicación diseñada para ayudar a configurar un sistema GNU/Linux para usar una conexión dial-up para el Internet. WvDial es útil para los usuarios porque puede detectar el modem y sólo requiere del usuario colocar el login de usuario, la clave y el número de teléfono que se necesita para conectarse. Una característica de WvDial es que no requiere de ningún script.

Se necesitan algunos pasos para configurar y ejecutar WvDial. Primero, editar el archivo wvdial.conf localizado en /etc/. En el archivo wvdial.conf, el usuario necesita editar el nombre de usuario, clave y número de teléfono para marcar. Después de editar este archivo, el usuario puede invocar el comando wvdial y la computadora marca al ISP del usuario.

Si se prefiere una aplicación GUI, el usuario puede usar las aplicaciones x-Wvdial o KWvDial que instalan las aplicaciones marcadoras usando una interfaz gráfica. La misma información que se necesita para el programa WvDial de línea de comandos se necesita para estos programas.

Para configurar el dial-up en estas aplicaciones, un usuario debe iniciar en File en el menu estándar, luego New y finalmente responder las preguntas. Una vez acabado, el usuario debe seleccionar conectar y entrar el nombre de usuario, clave y número a marcar.

Probando la Conectividad con ping

Una de las herramientas de troubleshooting más poderosas que puede usar cuando se configuran las interfases de red es el comando ping. El comando ping envía un paquete IP de información y espera por una respuesta. Si la respuesta es recibida, ya se sabe que la red y la interfaz están trabajando. Cuando se usa ping para examinar una interfaz, debe ejecutarlo desde la misma sub-red para que los problemas con el enrutamiento no afecten el resultado.

El ping funciona enviando un mensaje ICMP tipo 8 al host especificado. Cuando el destinatario recibe el mensaje, le responde con otro mensaje ICMP tipo 0. Si la red esta trabajando entre los dos hosts, el ping deberá ser retornado. Si hay problemas de red entre los host, el ping no será retornado. El ping puede ser invocado desde la consola con el siguiente comando: **$ ping host.dominio**

Ejercicio 2-7: ifconfig

Este ejercicio demuestra el uso del comando ifconfig, el cual es usado para ver y configurar las interfases de red. Debes ingresar como root para completar este ejercicio. Las soluciones para este ejercicio están proveídas en el Apéndice B.

1. Mostrar la configuración actual.
2. Mostrar la configuración actual de la primera tarjeta Ethernet.
3. Escribe debajo tu dirección IP y mascara de red.
4. Desactive la interfaz ethernet primaria.
5. Cambia tu dirección IP a 10.10.10.10, con una mascara de red de 255.255.0.0 y activa la interfaz.

6, Muestra los ajustes en la interfaz para que sea segura y correcta.

7. Adhiera un alias a la interfaz de 10.10.10.9 netmask 255.255.0.0.

8. Visualice la tabla de enrutamiento.

9. Restaura su dirección IP original. Reinicia si tienes problemas para que su conexión a la red trabaje nuevamente.

Ejercicio 2-8: tcpdump

Este ejercicio da un repaso completo a través del uso del utilitario tcpdump, el cual permite ver los paquetes de la red y como viajan a través de la red. Para hacer este ejercicio, necesitarás iniciar como root y tener la tarjeta de red en modo promiscuo. No se proveen soluciones para este ejercicio.

1. Muestra la configuración actual de tu tarjeta de red:

    \# ifconfig ethO

2. Ejecute el comando tcpdump sin opciones:

    \# tcpdump

3. Observe la información de cada paquete. La mayoría usted no la entenderá, pero podrá descifrar algunas líneas. Si no existe tráfico en la red deberá generarlo desde otra computadora en la subred.

4. Cambiese a otra consola virtual o xterm. Muestra su tarjeta de red de nuevo:
   # ifconfig eth0

Esta sentencia le notificará que la interfaz de red tiene la bandera PROMISC activada. Esto es porque tcpdump le dice a la tarjeta de red que quiere ver paquetes destinados para cualquier nodo en la sub-red, no sólo los paquetes direccionados a la dirección MAC de el sistema en el que usted esta.

5. Vuelva a la consola original. Detenga a tcpdump con CONTROL+C. Inicialo otra vez con el siguiente comando:
   # tcpdump -x -v

6. Muestre la salida de nuevo. Fijese que hay más información mostrada acerca de cada paquete. La opción -x descarga la cabecera en su código hexadecimal y la opción -v le dice que muestre todo el progreso en pantalla imprimiendo más información acerca de cada paquete. Detenga el programa, después de haber visto varios paquetes.

7- El programa tcpdump viene con un lenguaje de filtrado que se puede usar para mostrar ciertos paquetes. Se puede filtrar por dirección de origen y destino, número de puerto, protocolo, tamaño del paquete y varios otros ajustes. Revise las páginas man para más detalles. Ejecute el comando mostrando solamente los paquetes ICMP:
   # if tcpdump -x icrap

8. Efectué un ping a una máquina desde otra máquina en la misma subred:
   # ping -c 1 sistemaB

Todo lo que deberá ver es una petición ICMP y una respuesta. El protocolo que el comando ping usa.

9. Use el siguiente comando ping:
   # ping -p abcd1234 -c 1 sistemaB

Fijese que usted puede ver el abcd1234 en los paquetes ICMP.

10. Usted puede filtrar exactamente paquetes yendo desde un sistema a otro:
    # tcpdump src 192.168.0.1 and dst 192.168.0.2

11. Usted puede hasta filtro solamente tráfico Web:
    # tcpdump src 192.168.0.1 and dst 192,168.0.2 and port 80

12. Estudie a ver si puede idear otros filtros interesantes.

13. Si estas ejecutando el X, trate de ejecutar el programa ethereal. El provee un buen GUI para el programa tcpdump. El utiliza la mismas sintaxis de filtrado pero provee más información en un formato que es un poco más fácil para leer.

EL FUTURO - EL IPV6

Los profesionales del TI tenemos que siempre estar preparado para prever el futuro. Necesitamos una bola de cristal e intentar predecir los acontecimientos, para estimar como los próximos avances tecnológicos van a afectar nuestro trabajo y presupuestos. En este caso, la versión 6 de IP el IPv6, es el futuro y ya esta listo para escribirse.

En esta sección describimos la funcionalidad del protocolo IPv6 de la capa de Red. Se cubrirá el formato del datagrama y cabecera de IPv6, la fragmentación, como sistemas individuales son direccionados usando TCP/IP y algunas de los otros protocolos del TCP/IP que interactúan con IPv6, tales como arpa y el ICMP.

Los siguientes puntos son discutidos en esta sección:
- La necesidad del IPv6
- Repaso del IPv6
- Asignación de Direcciones
- La Transición hacia el IPv6

La necesidad del IPv6

El crecimiento del Internet es el simple y único factor que motivó la necesidad de desarrollar el IPv6. Las necesidades históricas de suplir espacio de dirección y asignación de direcciones IPv4 no ha resultado lo suficiente para llenar las futuras necedades del Internet.

Debido a este rápido crecimiento del Internet, el enrutamiento se ha convertido poco administrable. El mayor de los obstáculos es el direccionamiento. Las direcciones IP para poder suplir las demandas se están agotando debido al rápido crecimiento y al futuro crecimiento proyectado, y ya que hay un número finito de direcciones, lo que se quiere hacer incrementar el número para el futuro crecimiento del Internet. El IPv6 permite suficiente direcciones, para ser exacto son 4,096 direcciones por cada pie cuadrado del planeta tierra.

La proliferación de los palmtops, teléfonos celulares, la necesidad de aplicaciones QoS y la disponibilidad del Internet para negocios, y hasta los electrodomésticos que necesitan de direcciones IP es lo que han llevado y contribuido a esta necesidad de que el IPv4 evolucione al IPv6.

Mientras todavía hoy día el enfoque es en computadoras para la necesidad de direcciones IP en el futuro, más y más dispositivos serán conectados a las redes y al Internet. Esto llevará a una explosión en el número de direcciones requeridas por la población, esto sucederá aunque la populación de los usuarios del Internet vía navegación no creciese más, lo cual es imposible.

Repaso del IPv6

Para poder responder la necesidad de más direcciones IP, el RFC 1883 expandió el IP a la versión 6 (IPv6). El IPv6 empieza ofreciendo enrutamiento expandido y capacidad de direccionamiento mediante el uso de un espacio de dirección de 128-bit comparado con al espacio actual de direcciones de 32-bit. Esto permite simplificar la autoconfiguración de direcciones y a LANs usar una dirección MAC como la porción de host de una dirección para eliminar conflictos.

El IPv6 permite unicast, anycast y multicast pero no broadcast. Esto reducirá el flujo de tráfico y la sobre-cargas en el sistema y los procesadores de los routers. Además le ofrece un simple formato cabecera en el cual las tareas de cheksum son pasadas a las capas más elevadas del modelo OSI así para que los enrutadores y los switches se dediquen ha hacer sólo su trabajo.

El IPv6 ofrece soporte para opciones, capacidades de Calidad de Servicio (QoS), autentificación y capacidades de privacidad.

Para optimizar el flujo a través de los enrutadores, IPv6 ha reducido el número de campos en los cabezales del IP y ha introducido el concepto de extensiones de cabezales para manejar las funcionalidades menos usadas y para separar la información necesitada en los enrutadores de la que necesita en el host destinatario.

Los tamaños de los campos también han sido tomados en consideración para permitir alineamiento sobre fronteras naturales, lo cual facilita para que una parte del procesamiento se lleve acabo en el hardware.

El IPv6 no tiene checksum de cabezales; el cuenta con los protocolos encapsulados para llevar esto acabo, lo que significa que los enrutadores a lo largo de la ruta no tienen que implicarses con llevar acabo las calculaciones de checksum.

Las direcciones ahora son de 128-bits y pueden ser divididas en tres principales grupos por funciones:

Unicast	***Direcciona un único host***
Multicast	***Direcciona múltiples hosts simultáneamente***
Anycast	***Direcciona por lo menos uno de un número de hosts***

Los enrutadores que detectan que un datagrama anycast ha sido recibido no necesita remitir el datagrama. No hay broadcast ya que esta funcionalidad pueda ser proveída con multicasting a todos los hosts en una LAN.

Formato del Cabezal IPv6

El formato del cabezal del IPv6 es diferente del formato actual del cabezal del IPv4, como es ilustrado en la gráfica a continuación.

FOTOOOOOOOOOOOSSOOSOSOSOOSOOS

Los primeros bits tienen un significado predefinido:

Version	4-bit Protocolo de Internet número de versión igual a 6
Priority	4-bit Valor de Prioridad
Flow Label	24-bit Campo
Payload Length	16-bit Entero sin signo dando la longitud de la carga (Payload Length), ej: el resto del paquete después del cabezal IPv6 en octetos.
Next Header	8-bit Selector; identifica el tipo de cabezal inmediatamente después del cabezal IPv6; usa los mismos valores como el campo de protocolo IPv4.
Hop Limit	8-bit; Entero sin signo; disminuido por uno por cada nodo que reenvía el paquete; el paquete es descartado si el limite de salto (Hop Limit) se reduce a cero.
Source Address	128 bits; la dirección de quien envía inicialmente el paquete.
Destination Address	128 bits; la dirección de quien ha de recibir el paquete finalmente (posiblemente no el ultimo receptor si un enrutamiento opcional es presente)

La etiqueta de flujo (Flow Label) aún es experimental pero puede ser usada para identificar todos los

paquetes asociados con un flujo de información o conversación. Los enrutadores pueden hacer uso de esto para reducir el procesamiento de data para el flujo. Esto puede permitir decisiones hechas para el primer paquete en el flujo a ser mapiado rápidamente para toda la data subsecuente en el flujo.

Las direcciones IPv6 son escritas en hexadecimal con dos puntos separando cada dos octetos, de este modo, hay 8 piezas de 16-bits.

Asignación de Dirección

Con el el cambio del tiempo las asignaciones de IP han ido cambiando. El nuevo esquema de Asignación ocurrira en la version 6, como se muestra acontinuacion.

La actual Asignación de dirección IPv6 es como se muestra en la tabla.

Asignación	Prefix (Binario)	Fraccion
Reservado	00000000	1/256
Direcciones de red ISO	0000001	1/128
Direcciones de red Novell (IPX)	0000010	1/IZB
Provedores basados en dirección Unicast	010	1/8
Direcciones Unicast EOT reservadas basadas en geografía	100	1/8
Direcciones de uso de vinculo local	1111111010	1/1,024
Direcciones de Uso de Sitio local	million	1/1,024
Direcciones Multicast	nn mi	1/256

Nota que más del 70% aun esta siendo asignada!

Extensión de Cabezales

Las opciones será colocadas en un cabezal de extensión separadas. La mayor parte de los cabezales no serán procesados hasta que el paquete no llegue a su final de la destinación. Esto lleva acabo un incremento de velocidad por encima de la versión 4. Además, debido a la longitud arbitraria de la extensión del cabezal, futuras características adicionales podrán ser soportadas a medida que se conviertan necesarias.

Esta construcción permite más de lo siguiente:
- Enrutamiento Extendido (igual que IPv4 loose-source route)
- Fragmentación y reensamblamiento
- Seguridad de integridad y autentificación
- Confidencialidad
- Opciones especiales que requieren procesamiento de salto a salto (hop-by-hop)
- Información opcional a ser examinada por el nodo destinatario

Mejora para el soporte para extensiones y opciones incluyen lo siguiente:
- Cabezales Opcionales
- Flow Labeling- Método aún experimental de mantener opciones de punta-a-punta para conexiones mediante todos los enrutadores participantes:
    - Fragmentación sólo por el host de origen- Mínima MTU de 576 bytes en todos los vínculos
    - Debe soportar entrega de vínculo MTU.
    - Uso recomendado de PATH MTU Discovery (RFC 1981).
- Administración de ancho de banda- Campo prioritario de valor de 8 a 15 para data de tiempo real.
- Capacidades de autentificación y privacidad (RPCs 1826 y 1827)

ICMPv6

En ICMP, al igual como en otras partes de la versión 6, también experimentará cambios:

• Mensajes de error de ICMPv6:
    1 Destino no alcanzable
    2 Paquete demasiado grande
    3 Tiempo excedido
    4 Problema de parámetro

• Mensajes de información de ICMPv6:
    128 Responder petición
    129 Responder respuesta
    130 Consulta de Membrecías de Grupo
    131 Reporte de Membrecías Grupo
    132 Reducción de Membrecías Grupo

Integración de la Capas Físicas

Además hay un componente de integración de la Capa Física en el IPv6, ajustada en el RFC 1970. La Capa física será integrada lógicamente con la capa de Internet. Los siguientes servicios y puertos son cambiados:

- El Descubridor de Router es remplazado y mejorado por los puertos 133 (Solicitud de Router) y el 134 (Anunciador de Router).
- arpa es remplazado por los puertos 135 (Solicitud de Vecino) y el 136 (Anunciador de Vecino).
- Redireccionamiento de ICMP es colocado por el puerto 137 (Redirect).
- Un multicast especial formado es usado en lugar del broadcast arpa.

La Transición al IPv6

La transición al IPv6 no puede ser realizada con un simple script de implementación o en una rápida y única fase. La transición a IPv6 tiene que ser necesariamente lenta y metódica porque el Internet es demasiada amplia para poder efectuar estos tipos de cambios bruscos.

Debido a el inmenso número de servidores, enrutadores, redes, IPSs, ISPs, etc., que son parte del Internet, todos los usuarios tienen que habilitados para poder migrar individualmente en su propio tiempo y su propia forma. Es por esta razón que tiene que haber interoperatividad entre los dos el IPv6 y el IPv4, el estándar actual.

Ambientes Mixtos de IPv4 e IPv6

Dos formas especiales de direcciones son necesarias durante el período de transición:

- IPv4 Compatible, usadas en sistemas que puedan soportar IPv6.
  0000;0000:0000:0000:0000:0000:0102:0304 puede ser escrito ::0102:0304 o ::1.2.3.4
- IPv4 Mapiado, usado por sistemas que no soportan IPv6.
  0000;0000:0000:0000:0000:FFFF:0102:0304 puede ser escrito ::FFFF:0102:0304 o ::FFFF:1.23.4

Los formatos dados permiten protocolos de transporte puedan generar checksums correctamente en direcciones mapiadas. Si el que envía genera un checksum IPv4, el cual es más adelante mapiado, el recibidor sería capaz de verificarlo como un checksun IPv6 mapiado.

IPv4 Cliente, IPv6 Servidor

Durante el período de transición, habrán reglas especiales aplicados a todas las conversaciones en el Internet. En los casos donde un cliente IPv4 intente contactar un servidor IPv6, se aplicaran las siguientes reglas de contacto:

- El servidor es contactado por el cliente con una dirección IPv4. Los servidores deben tener dos direcciones mapiadas (una versión 4 y una 6) para poder responder las peticiones de los clientes desde ambos tipos de máquinas. El stack IPv4 mapea las direcciones, pero las capas superiores ven IPv6.
- Todos los sockets en el servidor son IPv6.
- Las reglas de contacto para las máquinas clientes que soportan IPv6 son mucho más simple; ¡Los clientes IPv6 simplemente usan IPv6!
- Sólo se espera del cliente IPv4 conozca IPv4 y buscará una dirección IPv4 para sus servidores.

Esta Búsqueda puede ser buena vía DNS- el cliente haría una petición de un record A donde los clientes IPv6 hicieron peticiones de records AAAA. Para que esto trabajo, el servidor debe tener una dirección IPv4 asignada; sin embargo, mientras que este esta preguntando internamente, una de las direcciones intranet será usada.

IPv6 Cliente, IPv4 Servidor

En situaciones donde el cliente soporta IPv6 y el servidor no, se aplican los siguientes reglas:
- El cliente contacta el servidor en una dirección IPv4 mapeada a una dirección IPv6.
- La Búsqueda de nombre de host retornaría direcciones mapiadas a hosts IPv6 demandando records AAAA para servidores IPv4.
- El stack de IPv4 mapea las direcciones.
- La capa superior ve IPv6.
- Todos los socket del cliente son IPv6.

Estos procesos son ilustrados en el siguiente diagrama.

(9999999999999999

Los clientes IPv4 simplemente usan IPv4 como ellos han hecho todo el tiempo.

El cliente IPv6 demanda una dirección AAAA para los servidores y el DNS contesta con una dirección IPv4 mapiada; esto causa el stack IPv4 a ser usados en comunicación con el servidor.

Tunneling Automático

Una de las más robustas características del IPv6 y una de las más interesantes, es el tunneling automático. Con el tunneling, por si no recuerda, es el proceso de establecer un canal seguro entre los nodos de Internet.

Los border routers en las fronteras de los sistemas IPv6 puede encapsular la data IPv6 dentro de un paquete IPv4 (estableciendo el campo del protocolo a 41). Si las direcciones usadas son compatibles con IPv4, entonces el paquete IPv4 sería enviado a la dirección IPv4. Esto es llamado tunneling automático.

Tunneling Configurado

IPv6 además soporta tunneling configurado. En contraste al tunneling automático, los border routers en cada extremo deberán saber información explicita de configuración. Si direcciones IPv4 compatibles no están disponibles, entonces los boundary routers necesitarían información de configuración explicita para suplir el gateway que los paquetes encapsulados son entregados, el enrutador destinatario. Esto es llamado tunneling configurado es simplemente pasar un tunnel IPv4 entre dos redes ejecutando IPv6.

Ejercicio 2-9: El IPv6 y el IPv4

No se proveen soluciones a este ejercicio.

Una aplicación ejecutándose en un cliente (A) implementando IPv6 envía data a un servidor (B) que implementa IPv4.

1.- Describa como la data avanza a través de la pila TCP/IP en ambos A y B y que dirección de destino es usada por B en cada etapa (no se preocupe por la dirección IP de A para este problema).

Ejercicio 2-10: Concerniente al IPv6

No se proveen soluciones a este ejercicio.

Es necesario proveer un reporte de estrategia de redes, y en particular un problema de asignación de dirección y como su COMPAÑÍA puede continuar proveyendo nuevas direcciones IP para los nuevos hosts durante la creciente expansión.

1.- La solución es obviamente migrar hacia el IPv6, ¿pero como?
Pruebe a ver si puede sugerir una estrategía que cubra los siguientes puntos:

A.- ¿En cuantas etapas por separado cambiaría usted por completo la red de la COMPAÑÍA?

B.- ¿En que orden cambiaría usted cada una de esta etapas?

C.- ¿Donde es lo más probable que necesitaría usted nuevo equipo/software?

D.- ¿Dejaría usted redes bajo el IPv4 en cualquier parte de la red de la COMPAÑÍA?

E.- ¿Cual otro beneficios derivados del IPv6 podría usted ofertar a los diferentes departamentos además de la no falta de direcciones IP?

F.- ¿Puede usted también tomar ventaja de las características del IPV6 para ofertar algunas mejoras potenciales en el futuro inmediato?

RESUMEN

En este capítulo, discutimos los siguientes conceptos básicos de redes:

- Como interactúan los diversos dispositivos de LAN/WAN, incluyendo los NICs, repetidores, hubs, bridges, routers, brouters, switches, gateways, CSU/DSU, modems, y los patch panels
- Las diferencias entre medios comunes de transmisión usados en redes, como son los cables par-trenzados, coaxial, y fibras ópticas y el medio inalámbrico (wireless)
- Los tipos de transmisión, incluyendo asincrónico y sincrónico, simplex, half duplex, full duplex, baseband, y broadband
- Las diferencias entre la topologías lógicas y física
- Los Estándares LAN, incluyendo los del IEEE (Instituto de Ingenieros Eléctricos y Electrónicos), Apple LocalTalk y FDDI (Fiber Distributed Data Interfase).
- Los conceptos fundamentales del direccionamiento IPv4 de 32-bit, incluyendo la estructura de la direc-

ción IP, clases de direcciones, reglas de direccionamiento y direcciones reservadas.
- Convertir direcciones binarias a formato decimal y al contrario.
- Crear mascaras de subnet para acomodar a las necesidades de una compañía y determinar los rangos de las direcciones IP de la subred.
- Las diferencias entre el IPv4 y el IPv6
- Estándares WAN, incluyendo el X.25, frame Relay, y Asynchronous Transfer Mode (ATM)
- Las funciones y los tipos de sistemas carriers T-carrier y E-carrier

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué archivo controla la configuración del inetd?

2. ¿Cuáles son las tres clases de direcciones IP?

3. ¿Qué hace el Protocolo de Resolución de Direcciones(arpa)?

4. ¿Qué son los tres componentes para la conexión del PPP?

CAPITULO 3

Redes del Sistema Operativo GNU/LINUX

ENRUTAR/ROUTING

Objetivos

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Estar familiarizado con las tareas de configuración- opciones de enrutamiento, redireccionamiento de IP (forwarding) y configuración del kernel- asociado con usar a GNU/Linux como un router.
- Describir los rol de ARP y las tablas de enrutamiento en la configuración de interfases de redes.
- Describir la instalación, el propósito y la implementación de firewalls.
- Describir la instalación e implementación de un servidor proxy.
- Describir el rol de broadcasting, asignación de direcciones y multicast en la configuración de interfases de redes.
- Describa los pasos involucrados en IP multicast así como las situaciones donde este método es útil.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué es una netmask (mascara de red)?

2. ¿Cómo es que funciona enrutamiento/routing?

3. ¿Por qué es que usamos enrutamiento Dinámico?

4. Dentro de una red Clase C, ¿Cómo podemos direccionar más de 255 máquinas?

5. ¿Cuándo es que un gateway se convierte en un firewall?

INTRODUCCIÓN

En este mundo interconectado por el Internet, el concepto de routing y packet forwarding juegan un papel importantísimo. No me puedo imaginar como se pudieran conectar todas estas computadoras y redes en el gran gigante que es el Internet sin la ayuda de los routers. Cada paquete que es emitido por cada computadora individual debe ser direccionado de alguna manera u otra para que pueda llegar a su destino. Cuando se da el caso y esta destinación no es dentro de la red LAN local, los routers se tienen que involucrar. En este capítulo, discutiremos como es que esto es llevado acabo y como es que esto funciona. Otra idea muy importante relacionada con enrutamiento es subnetting o subnetear. Ella permite que manejemos más efectivamente un grupo existente de direcciones IP y para facilitar conectividad entre segmento de la misma red pero construidas sobre diferente topologías. También cubriremos en una manera más breve la idea opuesta de subnetear que es supernetear/supernetting, la cual, por lo general, es para aliviar la carga sobre los routers del backbone, reduciendo el tamaño de la tablas de enrutamiento. Echaremos un vistazo al amplio tópico de filtrado de paquetes más específicamente al concepto de servidores de firewalls y proxy. Finalmente, repasaremos los conceptos más importante de enrutamiento Dinámico y los tipos de protocolos de enrutamiento.

REDES A SIMPLE VISTA

En los días de desarrollo de las redes nos encontrábamos con sistemas UNiX a larga distancias interconectados sobre líneas telefónicas estándar con varias instalaciones de líneas dedicadas para ellos. Estos sistemas usaban el protocolo TCP/IP para facilitar la comunicación. Con la integración de más y más computadoras en redes, TCP/IP se convirtió y continua siendo, el protocolo por defecto del Internet. Cuando nos referimos a “Internet” y “Internet” son dos cosas diferentes. El Internet es el nombre correcto de lo que se refiere como la conglomeración mundial de computadoras y redes. Una Internet se refiere a dos redes conectadas juntas.

Las siglas TCP/IP significan Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet). Cada parte del TCP/IP maneja un aspecto diferente del transporte de la data. El TCP maneja la diagramación (layout) de la data que se envía a través de la red asegurandose que el layout de la data enviada es recibida apropiadamente. El protocolo IP maneja el transporte de la data a su destino. Una analogía común es la de un documento enviado por el correo tradicional. El TCP representa la carta dentro del sobre, mientras que el IP representa la dirección escrita en el sobre.

IP es efectivo porque enruta los paquetes al computador receptor y a la red específica. Todo esto se logra usando una dirección IP de 32-bit en el cabezal del paquete. Interpretar un número de 32-bit no es un tarea fácil; la dirección es comúnmente anotada usando un formato de decimal separado por punto, Ej., 192.168.2.34. Pero claro esta, los humanos recuerdan nombres más fácil que recordar números. Debido a esto, existen librerías de funciones que proveen información de los hosts para nombres o números. Cual parte de la dirección denota un red o un nodo es determinado por la mascara. Por ejemplo, una dirección IP de 192.168.33.99 xon un mascara (netmask) de 255.255.0.0 denota una red de 192.168 y 33.99 es la computadora o host en particular en la red. Dependiendo de las necesidades del sitio, más direcciones pueden ser necesitadas; así pues que, el sitio puede implementar subnetting. Con subnetting, una dirección representa una red. Un computador asignan las direcciones IP internas a las computadoras en la red. Todas las peticiones son pasadas a través de un único host en la red interna al Internet y vice versa.

En esta próxima sección, discutiremos herramientas de red.

Herramientas de Red

GNU/Linux incluye varios utilitarios para diagnosticar, configurar y monitorear el estado de la red. Para más información, deberá consultar las páginas man de los comandos. Cubriremos cada uno de las siguientes herramientas más adelante en este libro:

- **ifconfig**

  Este es el comando usado para configurar el dispositivo de interfase de red. Con este comando, podemos asignar direcciones TCP/IP además de activar y desactivar el dispositivo. Si escribimos "ifconfig" sin ninguna opciones nos muestra la configuración de los dispositivos activos. Agregando la opción -a despliega el estado de todos los dispositivos, activos o inactivos.
- **netstat**

  Este comando muestra información sobre el estado de la red incluyendo las tablas de enrutamiento, conexiones y estadísticas de las interfases. Escrita en la línea de comando sin ninguna opción, nos despliega todos los sockets activos.
- **ping**

  El utilitario ping envía paquetes de data a un host en el sistema. Este comando ayuda en la determinación de la disponibilidad de un host en la red.
- **route**

  Sin opciones, el comando route nos muestra las tablas de enrutamiento. Este comando es usado para establecer direcciones TCP/IP estadística a otras máquinas.
- **traceroute**

  Este comando nos devuelve la ruta que un paquete toma a través del sistema y gateways en su camino al computador de destino.

SUBNETTING/SUBNETEAR

En redes TCP/IP, existe la necesidad de dividir redes en secciones. Las razones para segmentar un red incluyen proximidad de recursos, balanceado de carga de tráfico en la red, organización unidades de sistemas autónomos, o por otras variadas razones. Por estas razones y muchas un poco otras que están más allá del alcance de este libro, redes están en la necesidad de crear estas divisiones. En el ámbito de las redes TCP/IP, esto se lleva acabo con la implementación de subnetting.

Los siguientes tópicos son discutidos en esta sección:

- Conectar Redes
- Direcciones IP
- Definir Subredes/Subnetworks
- Ejemplos de Subnet
- Implementar una Subred
- Supernetting/Supernetear
- Classless Interdomain Routing/Enrutamiento de Interdominios sin Clase (CIDR)
- Consecuencias de las Tablas de Enrutamiento

Conectar Redes

Las redes pueden ser subdivididas en subredes lógicas. Desde el punto de vista del IP, los componentes de una dirección IP describen una única red. Esto no siempre es conveniente. Por ejemplo, considere una organización con varias LANs pequeñas, las cuales deben ser unidas para dar acceso al Internet.

En este caso, es posible subdividir una red de IP único en un número de subredes más pequeñas, llamadas subnetworks, donde cada una de estas, comparten la misma dirección IP de red. Para el mundo fuera de la red, sólo existe una red y toda la administración de enrutamiento, etc., puede ser manejada desde el nivel local.

Usando este mecanismo, también es posible subdividir los hosts en una red física en grupos lógicos que pueden ser considerados subredes. Este método es menos común pero puede también ser muy útil.

Direcciones IP

Una dirección IP es un número binario de 32-bit. Esto implica que se puede direccionar hasta 4 billones de hosts. Históricamente las redes Clase A, B, y C fueron libremente asignadas, y sólo una fracción de las direcciones en cada red son usadas. Esto resulto en un escaces de direcciones IP para las nuevas organizaciones. Unas solución de corto a mediano plazo fué el uso de subredes.

En la infancia del Internet nadie envisionó que esta creciese a los niveles que ha adquirido. Consecuentemente, compañías como IBM y la Hewlett-Packard que estuvieron en el desarrollo del Internet desde sus comienzos fueron otorgadas redes de Clase A completas.

La gran mayoría de estas organizaciones poseen un exceso de estas direcciones que fueron puestas a su disposición. Una sola red Clase B completa puede proveer 65,000 direcciones de IP para sus hosts. Muy poca organizaciones tienen m´s de unos cuantos miles de hosts en una red, así que un infimo porcentaje de las direcciones son usadas actualmente.

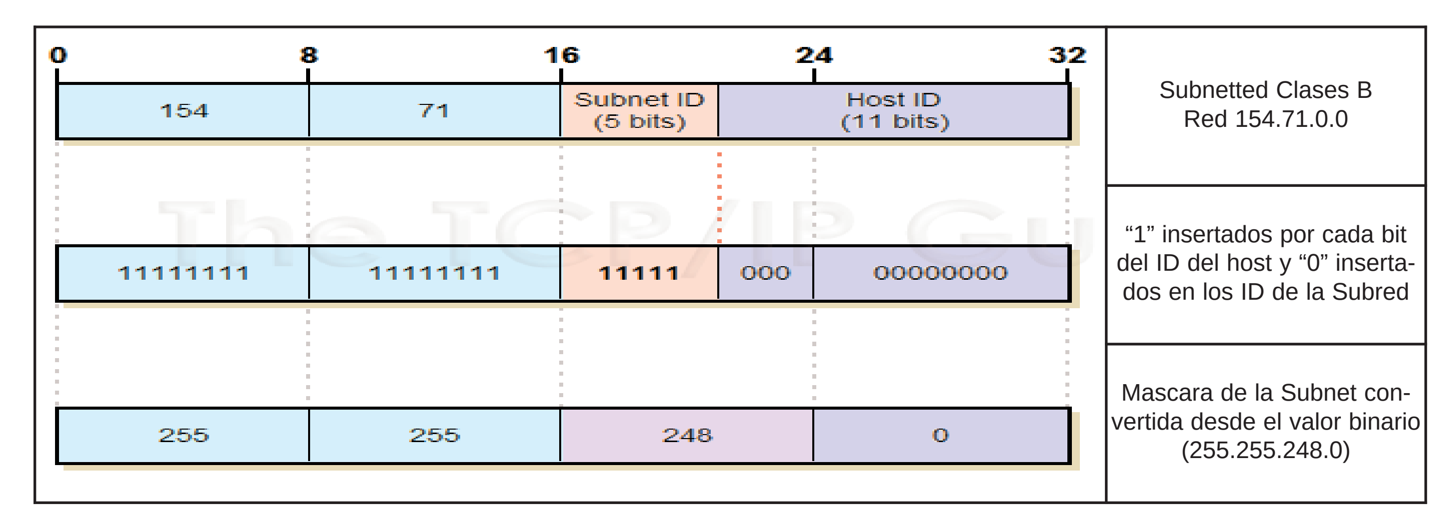
Como se mencionó anteriormente, una solución disponible es incrementar el número de bits en las direcciones IP a más de 32 así definiendo una nueva versión de IP, llamado el IPv6. Aunque esto no ayuda a las implementaciones de las ya existentes redes de IPv4.

Definir Subredes/Subnetworks

Es una tarea muy simple establecer subredes en su red d área local. Es llevado acabo con el uso de las mascaras de red/netmask. La netmask es una cadena de cuatro octetos de unos y ceros que es usada para enmascarar el componente de dirección de red en una dirección IP, así separandola de la porción host de la dirección. Esta cadena es especificada utilizando una anotación de decimales separados por punto. Por ejemplo:

255.255.0.0

Esta es la mascara por defecto de una dirección Clase B, donde los dos primeros octetos constituyen la dirección de red.



Con la manipulación de la mascara de red, es posible alterar la porción de la dirección IP que es usada para especificar un número de red. Es muy común, por ejemplo, usar el tercer octeto de una dirección de Clase B como un número de subred. Esto es bien fácil de establecer; simplemente se establece la mascara así:

255.255.255.0

Ahora es posible especificar hasta 254 subredes, cada una puede contener hasta 254 hosts y pueden operar independientemente. Esto es por lo general preferible a tener una red única con 65,534 hosts. Al mundo externo la organización aparenta ser una red singular con su número de una red Clase B.

En una situación donde no hace falta subnetear, entonces no se especifican subredes. Use la dirección IP de Clase B exactamente como es definida.

135.33.156.2	**Dirección IP Clase B**
135.33	**Dirección de Red**
156.2	**Número de Host**

En el siguiente ejemplo, usamos el tercer octeto completo para la dirección de subnet y el cuarto octeto para la porción de la dirección del host. No se necesita efectuar ningún tipo de recalculación.

- Use el primer octeto del número de host como el número de subred.
- Esto nos da 254 subredes con 254 hosts.
- Especifique una mascara de 255.255.255.0.

135.33.156.2	**Dirección IP Clase B**
135.33	**Dirección de Red**
156	**Número de Subred**
2	**Número de Host**

Aún cuando subneteamos, algunas direcciones de red mantienen su significado.

- El caso normal

135.33.156.2	**Dirección IP Clase B**
255.255.0.0	**Clase B mascara**
135.33	**Dirección de Red**
156.2	**Número de Host**
255.255	**Broadcast dirigido**

- Subnet case

135.33.156.2	**Dirección IP Clase B**
255.255.255.0	**Clase C mascara**
135.33	**Dirección de Red**
156	**Número de Subred**
255	**Broadcast dirigido a la Subred 156**
255.255	**Broadcast dirigido a todas las Subredes**

No podemos usar subredes cuando los números de las subredes tiene todos sus bits establecidos a uno. La razón para estos es que la dirección de broadcast en esta subred fuera la misma que la de broadcast para todas las subredes; así que la dirección de subnet con todos los unos es ilegal.

135.33.255.2	**Dirección IP Clase B**
255.255.255.0	**Mascara de Subred**
135.33	**Dirección de Red**
255	**Número de Subred**
255	**Broadcast dirigido a la Subred 255**
255.255	**Broadcast dirigido a todas las Subredes**

Un argumento similar nos detiene de usar la subnet con el número 0 (el 0 denota la ruta por defecto).

En general, se pierden dos números de subred (todos los unos y todos los ceros) cuando usamos subredes. Una excepción a esta regla es cuando usamos Classless Interdomain Routing (CIDR). Más adelante en este mismo capítulo discutiremos CIDR.

Ejemplo de Subred

Tomemos la siguiente dirección Clase B:

137.64.0.0

La mascara por defecto es 255.255.0.0. Esto nos devuelve 65,534 direcciones de host (descontamos la de todos los unos y la de todos los ceros).

La queremos dividir en por lo menos seis redes.

# of bits	Subnet Mask	# of Subnets	# of Hosts/Subnet
1	255.255.128.0	2(0)	32,766
2	255.255.192.0	4(21	16.382
3	255.255.224.0	8(6)	8,190

Una mascara de subred de 255.255.224.0 nos da ocho redes, de las cuales sólo seis son útiles.

137.64.0.0	137.64.32.0	137.64.64.0	137.64.96.0
137.64.128.0	137.64.160.0	137.64.192.0	137.64.224.0

Examinemos un caso más complejo de subnetear. Nuestros requerimiento es dividir una red única lógica en por lo menos seis redes (Ej., administración, ventas, nomina, soporte, producción y recursos humano). La mejor manera de hacer esto es encaminarse a incrementar la mascara del de la subred hasta que descubra cual mascara mejor cumple con sus requisitos.

La primera mascara a tratar es la de 1-bit, 255.255.128.0, la cual nos da dos posible redes, 137.64.0.0 y 137.64.128.0, de las cuales ninguna son válidas. Esto claro esta no satisface nuestro requerimiento, así que debemos extender la mascara.

Netmask	255.255.128.0	11111111	11111111	10000000	00000000
Network 0	137.64.0.0	10001001	01000000	00000000	00000000
Network 1	137.64.128.0	10001001	01000000	10000000	00000000

La segunda mascara a tratar es una de 2-bit, 255.255.192.0. Esta mascara nos da cuatro posible redes, dos de las cuales no son válidas (todos cero o todos uno), lo cual aún no llena nuestro requisito.

Netmask	255.255.192.0	11111111	11111111	11000000	00000000
Network 0	137.64.0.0	10001001	01000000	00000000	00000000
Network 1	137.64.64.0	10001001	01000000	01000000	00000000
Network 2	137.64.128.0	10001001	01000000	10000000	00000000
Network 3	137.64.192.0	10001001	01000000	11000000	00000000

Ahora probamos con una con mascara de 3-bit, 255.255.224.0. Esta nos da ocho posible redes, dos de las cuales son inválidas, lo cual si satisface nuestros requerimiento y nos sobran dos!

Netmask	255.255.224.0	11111111	11111111	11100000	00000000
Network 0	137.64.0.0	10001001	01000000	00000000	00000000
Network 1	137.64.32.0	10001001	01000000	00100000	00000000
Network 1	137.64.64.0	10001001	01000000	01000000	00000000
Network 3	137.64.96.0	10001001	01000000	01100000	00000000
Network 4	137.64.128.0	10001001	01000000	10000000	00000000
Network 5	137.64.160.0	10001001	01000000	10100000	00000000
Network 6	l37.64.192.0	10001001	01000000	11000000	00000000
Network 7	137.64.224.0	10001001	01000000	11100000	00000000

Ejemplo de Subnetear en una Clase C

La red Clase C 197.20.4.0 es mostrada aquí, subneteada en ocho subredes (se muestran sólo cuatro).

Subnet Mask = 255.255.255.224

Aquí podemos ver una red Clase C subneteada en 8(6) subredes. Donde sólo cuatro están siendo usadas en el diagrama en la imagen, lo que nos deja dos redes en caso de crecimiento o expansión.

Las direcciones de red tal vez le parezcan extrañas ya que no terminan en un cero. Esto es causa de que

la porción host de la dirección ahora solo es de 6 bits en longitud. Los dos bits de arriba del ultimo byte en la dirección ahora establecen el número de la subred, esto nos da las combinaciones 0, 32, 64, 96, 128, 160, 192, y 224, de los cuales 32, 64, 96, 128, 160, y 192 son subredes validas que podemos usar.

Ahora podemos subnetear cuanto sea necesario, mientras tenga suficiente direcciones de host en la subredes que estén en uso.

Implementando una Subred

Implementar una subred es un proceso que toma dos pasos. El primero, establecer la mascara de la subnet en su intefase. La mascara de subnet es usada para todas las rutas vía el interfase. Use el comando ifconfig o una herramienta gráfica equivalente para establecer la mascara del subnet. El segundo es, agregar sus entradas a la tabla de enrutamiento. Debe crear una entrada por separado para cada dirección de subnet usando el comando route o una herramienta gráfica equivalente.

Destination	Gateway	Flags	Refcnt	Use	Interfase
137.64.0	137.64.32.21	UG	2	8,989	eth0
137.64.32	137.64.32.3	U	12	256,448	eth0
137.64.64	137.64.32.21	UG	0	487	eth0
137.64.96	137.64.32.3	U	1	9,027	eth0
137.64.128	137.64.32.254	UG	1	942	eth0
....					
192.9.200	137.64.32.254	UG	1	96,196	eth0

En nuestro ejemplo, el host esta en la red 137.64, y hemos decidido que requerimos un mascara de subnet de 255.255.224.0. Si nuestra dirección de host es 137.64.32.3, esto significaría que estamos en el tercer host 0.3 en la subred 137.64.32. Para implementar una subnet, deberá imponerle la mascara de subnet que ha calculado a la interfase que maneja los paquetes de esa red. Debemos usar el comando apropiado para alterar la mascara de subnet asociada a la intefase. Tome en cuenta que la mascara de la subnet puede ser que no se le presente cuando despliegue la información del enrutamiento. El uso de la mascara es implícito debido a la ruta que esta siendo asociada con la interfase.

Una vez ha efectuado el cambio de la mascara de la subnet, usted podrá agregar rutas para cada subnet. Generalmente, estas rutas apuntarán a uno o más enrutadores en su subred local. Pero, usted puede si desea, tener varias subredes en el segmento que esta directamente conectado, aunque esto si trabaja en contra de los beneficios de subnetear.

Por lo general, todos los hosts en una red usaran la misma mascara de subnet. Ciertos softwares, como el RIP, pueden sentirse confuso si este no es el caso.

Super-netear/Super-netting

Donde Subnetear es la adición de más bits de redes desde nuestra disponibilidad de bits de hosts, supernetear es simplemente el inverso. Si VLSMs (Variable Length Subnet Masks) son soportadas, podemos entones acortar la mascaras así como alargarlas (RFCs 1517 hasta 1519). Ahora acortamos el número de bits asignado a la red.

Las tablas de enrutamiento pueden ser reducidas agrupando rutas con los prefijos similares debajo de una sola entrada en la tabla. La longitud del prefijo puede ser seleccionada para hacer así el grupo tan grande como sea posible. Todas las direcciones de destino com el mismo prefijo pueden compartir una misma ruta. Es útil principalmente no para configurar redes pero para usar en las tablas de enrutamiento para reemplazar bancos consecutivos de direcciones de redes con una sola dirección de red de un nivel más alto. Esto acorta la tabla de enrutamiento enormemente. En vez de una mascara de subnet para enmascarar la parte de red de

la dirección, usamos la llamada supermascara (supernet mask).

CIDR- Enrutamiento de Interdominio sin Clase (Classless Interdomain Routing)

CIDR hace uso de supernetting (RFCs 1517 hasta 1519). CIDR esta soportada por la versión 4 del Protocolo Border Gateway (BGP4) y es usada para reducir el tamaño de las tablas de enrutamiento en los routers del backbone. CIDR fué propuesto en el 1993, ya cuando las direcciones de redes habían sido repartidas con una medida de disponibilidad alrededor de todo el mundo. Rutas más específicas (con mascaras de red más largas) pueden ser usadas para sobrescribir el enrutamiento para redes asignadas en áreas geográficas incorrectas por el mé explicado anterior de disponibilidad en vez de áreas geográficas. El CIDR requiere que las direcciones de redes asignadas en el futuro sean en base a sus área geográfica. La IANA (The Internet Assigned Numbers Authority) distribuye bloques de direcciones a centros regionales. El centro regional entrega bloques de direcciones IP a los ISPs. La entidad ARIN (American Registry for Internet Numbers) administra a los Estados Unidos y, aunque su nombre insinúa diferente, a los otros paises (esta área de asignación de números antes era manejada por el InterNIC).

Consecuencias de las Tablas de Enrutamiento

Asumamos que un ISP desea asignar dieciséis redes Clase C a un cliente. Bajo condiciones normal esta situación requeriría dieciséis rutas. Si las redes tienen números consecutivos, sólo una ruta es requerida al usar supernetting.

FOTO FOTO FOTO=====

Aquí podemos apreciar como el uso de supernetear en CIDR es una gran ayuda para los routers backbone. Como este método puede drásticamente ayudar a reducir el tamaño de la tabla de enrutamiento. Pero si vale la pena mencionar que depende en la disponibilidad de bloques de direcciones consecutivas.

Esfuerzos a grande escala están siendo llevados acabo para reorganizar la colocación de direcciones de IP. Una de las ideas básicas es la asignación de conjuntos de direcciones Clase B a diferentes países y así simplificar la tarea de enrutar entre naciones.

Ejercicio 3-1: Subnetting/Subnetear

En este ejercicio, no necesitamos estaciones de trabajo. No se proveen soluciones a este ejercicio.

Tomemos las siguiente información en consideración antes de subnetear en ambiente de una compañía X. Esta compañía a decidido expandir y está en la necesidad de crear nuevos departamentos que necesitan ser interconectados a la red.

En la actualidad hay dos departamentos usando la siguiente configuración:

- Depto 1 usa 201.40.25.0 red de Clase C y tiene suficiente direcciones para hasta 20 hosts.
- Depto 2 usa201.40.26.0 red de Clase C que tiene suficiente direcciones para hasta 35 hosts.

Deseamos agregar dos Departamento más: uno que tiene 15 hosts y el otro con 35 hosts.

El único problema es que no tenemos más redes Clase C disponible para usarlas en estos nuevos departamentos. Tendremos que emplear una solución de subneteo para tomar las Clase C ya existentes y dividirlas para acomodar los nuevos hosts.

Además, debemos planificar cambiar para sólo usar una red Clase C única ya que en el futuro necesitaremos más redes para otros departamentos.

Deberá planificarse para:

- El cambio al nuevo esquema de subredes en cuatro departmento usando dos redes Clase C.

• El cambio a cuatro departamento con una red única Clase C.

Detalle cual departamento usará cual red, la mascara de la subnet a usar y cuantos hosts tenemos disponibles y con que rango de direcciones IP. Tome en cuenta cualquier router que desearía usar entre los departamentos para llegar a sus conclusiones.

CONCEPTOS DE ENRUTAMIENTO

Un entendimiento de los conceptos de enrutamiento es critico para su entendimiento de enrutar como ciencia.

Los siguientes tópicos son discutidos en esta sección:
- Enrutamiento de Red
- Interconectar Redes
- Envío de Paquetes IP /IP Packet Delivery (Red Local)
- Enrutador o Pasarela/Gateway
- Envío en Redes Multiple
- Tabla de Enrutamiento IP
- Orden de Búsqueda en la Tabla de Enrutamiento
- Desplegar la Tabla de Enrutamiento
- Establecer las Rutas
- Filtrado de Paquetes
- Servidor Proxy de SOCKS
- Comandos route y traceroute
- Modo Promiscuo

Enrutamiento de Red

¿Qué es enrutar? Los paquetes tienen que ser entregados desde una computadora a otra a través de las redes. El destino puede ser una máquina local o una remota. En cualquier de los casos, los paquetes tienen que ser enrutados desde la máquina fuente hacia la máquina destino.

Igual que el departamento de clasificación de la Oficina de Correo

Enrutar es el corazón de las redes modernas de TCP/IP. Podemos compararlo con el modelo de enviar una carta a través del sistema postal. Si deseamos enviar una carta, se nos ofrecen en esencia dos maneras:
- Si la carta es para alguien en su misma calle, usted puede caminar y colocarla en su buzón personalmente.
- Si está fuera de su calle, la coloca en el buzón de salida de la calle.

Las misma dos opciones se presentan a un computador en una LAN. Esta puede o entregar el paquete directamente a las máquinas en la red local o entregarla al enrutador que sabe como obtener la información acerca de la dirección de destino.

Interconectar Redes

Recuerde que todos los hosts en una red deberán tener una dirección IP única. La dirección contiene los componentes:
- La parte Red
- La parte Host

El componente de la dirección que es red será igual para todos los hosts en una red en particular. así que, en este diagrama tos los hosts en la red superior tienen una dirección IP con el componente de red de 192.9.100. Similarmente todos los hosts en la red inferior tienen dirección IP con el componente de red

de192.9.200.

Para poderse comunicar entre host A (192.9.100.2) y host B (192.9.200.4), deberá existir una manera de transferir los paquetes IP desde la red192.9.100 a la red 192.9.200 y vice versa.

Envío de Paquetes IP (Red Local)

En el siguiente ejemplo:

- Usted esta en el punto A.
- Usted mismo puede entregar una carta al punto B, ya que B le queda en la misma calle.
- Usted entonces no usa el buzón de salida.

Entregarle correspondencia a alguien en su misma calle es una tarea simple, y como es costumbre no involucra el correo oficial. Este es el mismo proceso que ocurre cuando se transfiere data entre dos computadoras en la misma red local.

La tarea de reenviar un paquete hacia su destino final es manejado por el IP sin el conocimiento de los protocolos de más alto nivel de transporte.

En un caso simplificado, donde el paquete esta siendo enviado a un sistema en la misma red, el IP simplemente pasa el paquete a la capa de Enlace de Data, dirigiendola hacia su destino. En este caso, las tramas de enlace de data (data-link) y los datagramas de IP son direccionados a mismo sistema de destino. En el ejemplo anterior, A le da el paquete IP a la interfase de red. Entonces, la interfase de red coloca el paquete IP en una trama Ethernet direccionada a B. Luego, la trama Ethernet es enviada en la red local. Aquí B recoge la trama Ethernet con su dirección de destino y extrae el paquete IP, entonces el paquete IP se envía para arriba hacia la capa de Transporte.

Enrutador o Pasarela

La terminología varia, pero en realidad no existe la gran diferencia, entre un enrutador (router) y una pasarela (gateway). Históricamente, un gateway se refiere a una computadora con capacidades de enrutamiento habilitadas, mientras que un router es un dispositivo dedicado físico. Pero gateway/router conectan dos o más redes. así que, este tiene más de una interfase de red y respectivamente más de una dirección IP. El gateway/router debe ser visible en ambas redes y cada dirección IP esta asociada con la interfase de la red. Los host que cumplen esta condición a menudo también son referidos como hosts multihomed.

Network A:192.9.100 Host-1: 192.9.100.99 **Network B:192.9.200 • Host-2:192.9.200.99**

En primera instancia estas dos redes deberán estar conectadas físicamente. Esto se logrará usando un equipo que tendrá dos interfases y estará conectado a ambas redes. Tal sistema es conocido como un gateway.

Además de estar físicamente conectado a las dos redes, el gateway debe poder ser direccionado desde ambas redes. En pocas palabras, Este debe parecer como un sistema normal de la red 192.9.100 y también de la red 192.9.200. Para esto ser posible, el gateway deberá tener dos direcciones IP, una que es válida en la red 192.9.100 y la otra que sea válida en la red 192.9.200.

Las direcciones IP son asociadas con las interfases de redes apropiadamente conectadas a la red y esto significa que el gateway puede comunicarse con ambas redes.

Envío en Multiple Redes

En el siguiente ejemplo:

- Usted entrega una carta a su oficina de correo en el buzón local, porque B esta en una calle diferente a la suya.
- El servicio postal la recoge y la lleva oficina de clasificación.
- Entonces es entregada a B.

En este ejemplo no estamos enviando correo a alguien que vive en nuestra misma calle. El problema desde nuestro punto de vista, no es más complejo. Nosotros simplemente ponemos la carta en el buzón local y es todo.

El servicio postal llega, recoge la carta desde el buzón local, lo toma para la oficina de clasificación. Aquí la dirección escrita en la carta es usada para decidir donde se debe enviarla. Esta puede ser enviada a otra oficina para aún más clasificación, dependiendo que tan lejos esta deberá viajar.

Al la carta llegará a su destino B entregada por el servicio postal.

Este mismo proceso también es el que es aplicado a los paquetes de una red de computadoras para la entrega de paquetes.

A	C	B

Fijese que la dirección de la capa de Internet no ha cambiado- la carta aún es dirigida a B- pero el emisor primero tiene que entregarle la carta a un tercero,la oficina de correo.

Si el destino se encuentra en una red diferente a la de la fuente, el IP debe reenviar el paquete hacia el sistema que actúa como un gateway en esa red. Para hacer esto, el datagrama de IP es encapsulado en una trama de enlace de data con destino al equipo gateway. Cuando el equipo recibe el datagrama, se da cuenta que realmente no esta destinado para el propiamente y entonces procede a usar su propia tabla de enrutamiento para ver si lo puede reenviar a su destino. Esto puede involucrar enviarlo directamente a su destinatario en la misma red que el proxy o quizás enviarlo más adelante a otro gateway si aún se encuentra en una etapa intermedia.

```
# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2       192.168.2.7     255.255.255.0   U     0      0        0 eth0
200.42.200.0    200.42.100.1    255.255.255.0   U     0      0        0 eth1
196.3.81        192.168.2.7     255.255.255.0   U     0      0        0 lo
```

En este ejemplo, el paquete necesita viajar por el router (C):

1.- A sabe que B no esta en la Red 1, así que envía el paquete IP en una trama Ethernet direccionado a C, a la dirección por defecto para los paquetes dirigidos a los hosts, A no sabe esto ya que esto es función interna de C.

2.- C extrae el paquete IP desde la trama Ethernet, lo inspecciona, y descubre que su destino final no es C.

3.- C es local a ambas A y B, y por esto puede colocar el paquete IP en otra trama Ethernet en la Red 2 y la direcciona a B. Note como C tiene dos direcciones de diferente Ethernet en este caso- una en la Red 1 y otra en la red 2.

4.- B recibe la trama de Ethernet que proviene de C, le extrae el paquete IP, y lo envía hacia arriba a la capa de Transporte.

Tabla de Enrutamiento IP

EL IP toma su decisión en enrutar paquetes usando una estructura de data llamada tabla de enrutamiento (routing table). El formato de su tabla de enrutamiento depende de cual sistema usted tiene en uso.

La tabla de enrutamiento contiene las entradas para cualquier red que el sistema local tiene conocimiento. Este da la dirección del sistema de gateways para cada una de las redes que desea conectarse. El gateway debe estar físicamente conectado a la misma red como el host local. La entrada también específica cual interfase de red deberá ser usada para enviar los paquetes al gateway.

Red 1 128.6.4
Red 2 128.6.21
Red 3 128.121
128.6.4.2 ordenador A
128.6.4.3 ordenador B
128.6.4.1 128.6.21.2 gateway R
128.6.21.1 128.121.50.1 gateway S
128.121.50.2 ordenador C

La primera línea muestra una entrada para la red local. En este caso, el gateway esta en el sistema local. En la segunda línea nos muestra la ruta desde la 200.42.100 a la red 200.42.200. En este caso, el gateway esta físicamente conectado a la red de destino. La tercera línea contiene la ruta a una red que no tiene el gateway conectado directo a ella, o sea a ambas 192.168.2 y 196.3.81. En este caso, la tabla de enrutamiento contiene la dirección del gateway que lo lleva a la próxima etapa de la ruta.

Un Ejemplo de Enrutamiento

El computador A puede enviar datagramas al B directamente, usando la red 1. Sin embargo, no puede llegar al ordenador C directamente, puesto que no están en la misma red. Hay varias maneras de conectar redes. En el gráfico asumimos el uso de gateways (existen otras alternativas). En este caso, los datagramas que van desde A a C deben ser enviados a través del gateway R, red 2 y gateway S. Todos los ordenadores que usan TCP/IP necesitan que se les suministre la información y algoritmos apropiados para que puedan saber cuándo un datagrama debe ser enviado a través de un gateway, y elegir el gateway apropiado.

Tabla de enrutamiento de A:

A es local a B y R por esto puede entregar el paquete directamente a B. Pero no es local a C por eso lo envía vía R.

Tabla de enrutamiento de B:

B es local a ambos A y R, por eso entrega paquetes directamente a ambos. Para llegar a A, esta lo entre directo.

Tabla de enrutamiento de C:

C es local al host S. así que, este no puede entregar paquetes a ningún host directamente que no sea S.

Tabla de enrutamiento de R:

R es local a A y B y a S y por eso puede entregar paquetes directamente a ellos. Este no es local a C, así que este puede enviar paquetes para C vía S.

Tabla de enrutamiento de S:

S es local a C y a R y por eso puede entregar paquetes directamente a ellos. Este no es local a A ni a B, así que este puede enviar paquetes para A y B vía R.

La tabla de enrutamiento para un host 21.30.20.0 con router 21.30.20.0 a otras redes	Destino	Ruta	Tipo de Ruta
	21.30.20.0	Entrega Directa	Network
	136.125.1.0	Via 21.30.20.1	Network
	198.224.3.1	Via 21.30.20.1	Host
	198.224.3.2	Via 21.30.20.1	Host
	201.12.30.0	Via 21.30.20.1	Network

Rutas a Redes

Para grandes redes, especificar la dirección IP de cada host en la tabla de enrutamiento se convirtiera en inmanejable debido a su tamaño. Imaginese, que, cada dirección IP en el Internet tenga que ser incluido en una tabla de enrutamiento. La única solución es usar rutas a las redes. Estas son simple direcciones de red en una tabla de enrutamiento.

La tabla de enrutamiento para un host 21.30.20.0 con router 21.30.20.0 a otras redes	Destino	Ruta	Tipo de Ruta
	21.30.20.0	Entrega Directa	Network
	0.0.0.0	Via 21.30.20.1	Default

No es nuestro interés tener que especificar una ruta a todas las diferentes redes en el mundo. Si existe un paquete con una dirección de destino que la tabla de enrutamiento no contiene una ruta especificada, usaría la ruta por defecto. Es por lo normal la última entrada en la tabla de enrutamiento. La dirección IP 0.0.0.0 se refiere a la ruta por defecto en la tabla de enrutamiento.

Rutas Hosts	Mascara más larga
Rutas Redes	Mascara más Corta
Ruta por Defecto	

La rutas por defecto agregan otro nivel de simplificación a tablas de enrutamiento. Si no existiese tal cosa como la ruta por defecto, necesitaríamos especificar la ruta a todas las redes que desearíamos contactar en cualquier momento- imaginese todas las redes en Internet.

Orden de Búsqueda en Tabla de Enrutamiento

El orden en el cual la tabla de enrutamiento es buscada es de extrema importancia. La búsqueda siempre se efectúa desde la más específica hacia la menos específica.

Esto es controlado por la mascara para la ruta en la tabla de enrutamiento. Mascaras más largas, esto quiere decir más específicas, son igualadas con preferencias comparadas con las más cortas o menos específicas.

Destination	Route	Type	Metric
127.0.0.1	127.0.0.1	Host	0
21.30.20.0	21.30.20.10	Network	1
21.30.20.0	21.30.20.20	Network	2
0.0.0.0	21.30.20.1	Default	5

Por ejemplo la segunda ruta con el metric más pequeño (1) será usada ya que es la más rápida.
La ruta con el metric 2 será usada sólo si el hosts 21.30.20.10 no está disponible

Aprenderemos más sobre las más caras más adelante, pero por ahora, mantenga presente que al final es la mascara de la ruta que es usada para determinar la igualdad en la tabla de enrutamiento, no se asume nada acerca de la clase o tipo de rutas involucradas.

Metrics

Metric es una medida de cuantos gateways/enrutadores (hops/saltos) un determinado paquete tendrá que atravesar en su viaje para arribar a su destino. Esta medida es usada para elegir una ruta donde existe más de una opción disponible. Esta ayuda para determinar cual ruta es la más corta para llegar al destino de para entreg el paquete.

Un metric no tiene que realmente reflejar el verdadero número de saltos involucrado en una ruta. Es simplemente un parámetro que puede ser tan arbitrario como sea necesario.

Es muy útil diseñar que que se pueda elegir entre diferentes posibilidades de enrutamiento, quiero decir, donde proveemos más de una ruta para entregar un paquete dado. Al configurar multiple rutas a un mismo destino, los paquetes pueden ser entregados aunque una de las rutas este temporalmente fuera de servicios por cualquier razón.

Fijese en en esta tabla anterior, la ruta por defecto ha sido arbitrariamente establecida con un metric de valor 5; aunque este host esta conectado al Internet, el número real de saltos que un paquete tendrá que efectuar para llegar a su destino pueden ser hasta 30.

Desplegar la Tabla de Enrutamiento

Para desplegar la tabla de enrutamiento desde la línea de comando, podemos usar el comando route o el comando netstat con la opción -r.

```
# route
kernel   IP routing Table
Destination        Gateway       Genmask            Flags   Metric   Ref   Use   Iface
255.255.255.255    *             255.255.255.255    UH      0        0     0     eth0
192.168.1.0        *             255.255.255.0      U       0        0     0     eth0
127.0.0.0          *             255.0.0.0          U       0        0     0     lo
default            tricom.asdl   0.0.0.0            UG      1        0     0     eth0
# netstat  -r
kernel   IP routing Table
Destination        Gateway       Genmask            Flags   MSS   Window   irtt   Iface
255.255.255.255    *             255.255.255.255    UH      0     0        0      eth0
```

192.168.1.0	*****	**255.255.255.0**	**U**	**0**	**0**	**0**	**eth0**
127.0.0.0	*****	**255.0.0.0**	**U**	**0**	**0**	**0**	**lo**
default	**tricom.asdl**	**0.0.0.0**	**UG**	**1**	**0**	**0**	**eth0**

En GNU/Linux (y algunos otros sistemas operativos), usted puede usar el comando netstat para ver la tabla de enrutamiento. Los detalles de esto se cubren en los ejercicios asociados con este capítulo. Muchos de los comando de redes son iguales o muy similares a los de todas las distribuciones de GNU/Linux o los diferentes sabores de UNiX.

Estableciendo las Rutas

La tabla de enrutamiento pueden ser mantenidas manualmente por el administrador de sistemas usando el comando route. Este comando es mayormente usado para inicializar un sistema con la ruta a gateway por defecto en la red. El sintaxis de este comandos es:

route comando destino [netmask] [gateway] [metric] [dispositivo]

En este sentencia donde comando es uno de dos agregar o eliminar (add o delete) destino es la dirección IP a entregar el paquete, netmask es la mascara de la ruta el gateway es la dirección de la pasarela, metric es para elegir entre diferentes gateways con el mismo destino y dispositivo es para elegir entre un dispositivo y otro. Aquí le presentamos unos cuantos ejemplos que deben aclarar esto un poco:

```
route add -net 192.166.1.0 netmask 255.255.255.0 eth1
route add -net 10.10.0.0 netmask 255.255.255.0 eth2
route add default eth0
```

Enrutar es un Camino de dos Vias

Uno de los problemas más comunes de enrutamiento al configurar las tablas de enrutamiento es entender que debe ser efectuado en cada punto. Un escenario es Máquina A trata de hacerle ping a Máquina B. A ha sido configurada correctamente para enviar los paquetes de B a través del gateway R. R también ha sido configurada para enviar los paquetes entre las dos redes y entregar tanto a A como a B. Pero, B no no tiene en su tabla de enrutamiento como entregar los paquetes a A. así pues que no sabe como responder la petición del comando ping entrante de A. así que A no recibirá ninguna respuesta de su ping o cualquier otro servicio de red porque N no sabe como enviarle los paquetes de respuesta.

La solución es decirle a B que envíe los paquetes de A vía el router R. Luego los paquetes podrán fluir apropiadamente en ambas direcciones y el ping funcionará.

Filtrado de Paquete

Filtrado de paquete controla cuales paquetes IP son permitidos arribar a ciertas direcciones. Puede ser usado para limitar acceso foráneo a las computadoras locales y vice versa. En GNU/Linux, el filtrado de paquetes se controla a nivel del kernel. Existen módulos para el kernel que permiten definir un sistema de reglas para aceptar o rechazar los paquetes o las comunicaciones que pasan por el sistema. Estos sistemas de reglas conforman lo que se conoce como firewall o cortafuegos; en otros sistemas los firewall pueden estar implementados en software y estar desvinculados del sistema operativo, pero en el caso de linux, el firewall se puede montar a nivel de kernel y no es necesario instalar un software adicional que para más a veces tiene agujeros.

La gran mayoría de los servicios de Internet son almacenar y enviar (como es el correo y las noticias) o interactivos (como el FTP, WWW):

- El firewall normalmente será un host de reenvío para los servicios de almacenar-enviar. por ejemplo, el

correo se le envía al firewall, el cual lo reenvía hacia la red cual el protege.
- Son los servicios interactivos que requieren del uso explicito de proxies.

Las ipchains

En versiónes de Kernel anteriores a la 2.4 se disponía de los módulos IPCHAINS para montar firewalls, así como de otros módulos de soporte para comunicaciones concretas (enmascaramiento de FTP, de IRC, Real Audio etc.).

Gracias a IPCHAINS, además de poder definir normas de firewall muy precisas, también podía hacer que un linux funcionara como gateway enmascarando todas las peticiones de una LAN.

El filtrado de paquetes es proveído al nivel del Kernel en GNU/Linux por la herramienta ipchains (versión del kernel 2.2 y anteriores) y iptables (2.4 en adelante), ipchains funciona revisando reglas para determinar que hacer con determinado paquete. Tres chains son internas (built in): input, output, y forward. La cadena de entrada (input chain) es revisada cuando un paquete arriba. Si un paquete va hacia otra máquina, la cadena de reenvío (forward chain) es revisada. Justamente antes de que un paquete salga, la cadena de salida (output chain) es revisada.

Manejabilidad, los usuarios pueden definir sus propias cadenas y agruparlas con cadenas ya existentes. Por ejemplo, si el firewall se conecta al computador local vía el dispositivo eth0 y al Internet vía el dispositivo ppp0, uno puede definir las cadenas ppp-out, ppp-in eth-out y eth-in y agruparlas con las cadenas correspondientes.

Para definir una nueva cadena use el comando ipchains -N nombre-cadena. Las reglas pueden ser insertadas o agregadas en una cadena con las opciones -I -A.

Firewalls

Las técnicas de conta-fuego o firewalls son las más comunes para proveer seguridad de redes. El tópico de conta-fuego es muy amplio, así que aquí solo cubriremos brevemente los temas involucrados.

La manera más simple de un firewall es un router con el reenvío de paquetes no disponible. La meta principal es proveer protección del mundo externo. En algunos casos el administrador de sistemas puede decidir restringir el acceso a los usuarios al mundo externo. Como resultado, el firewall incrementa el rendimiento de los servicios y el tráfico al limitar otros. Una desventaja obvia de los firewall es la manera en que ellas restringen tráfico que se dirige fuera de la red al igual que en el que entra.

Una máquina conectada a dos redes es llamada un multihome host (esto quiere decir que tiene homes en dos redes y puede ver ambas). Un firewall es simplemente un host multihomed que tiene la opción de reenvía de paquetes apagada. Con el reenvía de paquetes desabilitado, el firewall puede ver ambas redes puro los host en esas redes no pueden ver esos host en el otro lado del firewall. En vez de desabilitar todo el reenvío de paquetes algunos firewall desabilitan casi todos los paquetes pero permiten que algunos útiles pasen. Este firewall esta ejecutando funciones de enrutamiento y es conocida como un screening (filtrado) router. El firewall es diseñado para proteger una red del acceso a usuarios no deseados del exterior. Podemos verlo como un par de mecanismo: uno que bloquea tráfico y otro que lo permite. El énfasis en uno o el otro se determina por la política de seguridad.

El problema de poder asegurar cada hosts en una red es simplemente demasiado dificultoso y consume demasiado tiempo, claro dependiendo del tamaño de la red. Aunque es simple tarea asegurar quizás unos seis hosts en una red pequeña, no es el caso asegurar una red quizás con unos 2,200 hosts, esto simplemente fuese una pesadilla, especialmente que se estuviese que enfrentar la individualidad de cada host.

Es mucho más fácil y simple asegurar una red completa, que asegurar los clientes individuales. Para lograr esta situación, conectamos la red al mundo externo vía un firewall, el cual bloquea toda la comunicación entre las redes. Los hosts en cada red pueden ver sólo el firewall, pero no se pueden ver entre ellos. Aún debemos asegurar un host, claro esta el firewall- ya que este es visible al mudo fuera de nuestra red.

Servidores Proxy

Los firewalls son fantásticos en cuestión de seguridad, pero tienen una pequeña falla, y es que la red interna tampoco puede ver fuera de ellos! así que necesitamos poder selectivamente transmitir algunos paquetes IP a través del firewall. Un Servidor Proxy es exactamente lo que necesitamos. Ellos re-transmiten paquetes enviados fuera de la red pero solamente ciertos tipos de paquetes (e.j., WWW).

Los servidores proxys nos permiten comunicarnos a través de un firewall enviando solamente un pequeño conjunto de nuestros mensajes. Ellos permiten solamente un conjunto de transacciones, prohibiendo todo el otro tráfico. El beneficio de un proxy es que solamente permite que tome lugar un tipo de comunicación, por ejemplo peticiones FTP o WWW. así como canalizando todas las peticiones a través de una pequeña ventana el proxy permite que cada petición sea registrada en un diario a un más, por lo general los proxys son configurado para solo permitir peticione salientes, nunca entrantes. Fijece que los servidores proxys no incrementan la seguridad; sino que la reducen. Hay un mecanismo de proxy genérico llamado SOCKS (que significa sockets, lo cual discutiremos más adelante) que permite crear proxies para cualquier servicio de Internet.

Los Proxies Tradicionales

Con los proxies tradicionales, los paquetes que provienen del Internet nunca entran directamente la red local y los paquetes de la red local nunca entran directamente al Internet. La conexión al Internet solo se puede efectuar a través del firewall. Cualquier servicio que uno desea que sea accesible desde el Internet (como es un servidor Web) debe estar en el firewall. Una aplicación en la red local debe estar informado de la existencia de un proxy y conectarse explícitamente al proxy en el firewall. Esto puede ser problemático ya que algunos software no soportan el uso de servidores proxies y por esto, no funcionarían correctamente, si es que funciona. SOCKS es un ejemplo de un servidor proxy tradicional.

Proxies Transparentes

Los proxies transparentes no requieren que las aplicaciones sepan de su existencia. El proxy debe aún estar ejecutándose en el firewall, pero las aplicaciones simplemente se conectan como siempre y los paquetes son transparentemente redirigidos al proxy cuando pasan por el firewall. El Squid es un ejemplo de un servidor proxy que puede ser configurado para ser un servidor proxy transparente.

Podemos por ejemplo redireccionar conexiones salientes desde la red local hacia el proxy en el firewall usando ipchains. Por ejemplo:

```
# ipchains  -A input  -p tcp -s 192.1.2.0/24 -d / BO -j  REDIRECT 8080
```

La dirección IP de origen 192.1.2.0/24 es usada en este ejemplo. El 24 le dice a ipchains que solo los primeros 24 bits son importante ya que la red completa esta siendo referenciada y no un solo host. Esto redireccionará todo el tráfico desde la red local hacia el puerto 8080 en el firewall, donde un servidor proxy estará esperando.

Enmascarando (Masquerading) el IP

Los paquetes de la red local traviesan todo el Internet de punta a punta, pero sólo por tratamiento especial que reciben. Software de proxy no es necesaria si usamos enmascarado de IP, pero una utilidad

especial del kernel llamada enmascarando es usada. Los paquetes desde la red local en su ruta hacia el Internet son rescritos por el firewall para que así ellos parezcan haber originado desde el firewall. Los paquetes desde el Internet hacia la red local son también rescrito para que así ellos aparezcan haber haber originado en el servidor proxy y simplemente estén dirigidos hacia el computador cliente.

Mucho de los programas clientes no entiende a los proxies, así que no pueden ser usados. Esto incluyen muchos clientes FTP. Este problema se resuelve usando enmascaramiento de IP. La configuración de un proxy muchas veces depende de un navegador Web como es mozilla o Netscape para todos los requerimientos ya que ellos si entienden los proxies.

En el siguiente ejemplo creamos una cadena ppp-in, la soldamos a otra cadena input y establecemos las políticas por defecto del dispositivo ppp0 al valor de DENY, lo cual significa que solo específicas conexiones las cuales fueron habilitadas posteriormente en la configuración son permitidas:

```
# ipchains  -N ppp-in
# ipchains   -A  ppp-in   -j   DENY
# ipchains  -A input  -j  ppp0  -1  ppp0  -j ppp-in
```

En el siguiente ejemplo permitimos conexiones ser hechas al servidor web llamado http://www.portal.com:

```
# ipcliains  -A ppp-in  -p TCP  -d www.portal.com 80  -j ACCEPT
```

Antes de que enmascara del IP funcione, el reenvío de IP (IP forwarding) debe ser establecido ejecutando el comando echo 1>/proc/sys/net /ipv4/ip_forward. La dirección IP es asignada dinámicamente, como es PPP o SLIP, el comando echo 1 >/proc/sys/net/ipv4/ip-_forward debe ser ejecutado. Luego, asegurandose que la interfase externa se llame ppp0, ejecute el siguiente comando:

```
# ipchains -P forward DENY
# ipchains -A forward -i ppp0 -j MAS0
```

Estos pasos establecerán una configuración básica de enmascaramiento de IP.

Las iptables

IPCHAINS ha muerto, viva IPTABLES. A partir del kernel 2.4 se esta dando soporte a otro módulo para filtrado de paquetes mucho más potente que IPCHAINS, llamado IPTABLES. Para acceder a ciertos sites ftp tendremos problemas usando IPCHAINS con el kernel 2.4. A pesar de que IPCHAINS siga funcionando, ya no tendremos los antiguos módulos para solventar los problemas de acceso a servicios especiales y se recomienda pasarse a IPTABLES.

Diferencias respecto a IPCHAINS

IPTABLES es más completo que IPCHAINS, permite un control aún más preciso, aunque también es más complejo.

En principio el sistema sigue siendo el mismo. Hay que cargar un módulo del kernel (se puede hacer en el propio script de firewall), y ejecutar un script de shell convencional que tiene el aspecto de un conjunto de reglas. Un script de este tipo se podría complicar y sofisticar tanto como se deseara, eso ya es cuestión de cada cual.

El script generalmente: · Comienza cargando los módulos necesarios (los imprescindibles y los auxiliares, como el de ftp masquerading), · Establece algún bit como por ejemplo el de forwarding. · Luego borra todas las reglas actuales (flush). · Establece las políticas por defecto para la aceptación, reenvío y salida. · Y finalmente va aplicando todas las reglas de firewall, que varían enormemente dependiendo de las necesidades de cada red. El orden de algunos puntos no tiene por que ser siempre así.. Por lo general, una aproximación buena suele ser CERRAR todo por defecto, e ir abriendo lo que se necesite. Aunque esta última aproxi-

mación da mucho más quebraderos de cabeza.

Diferencias respecto a IPCHAINS.

- La sintaxis, obviamente, aunque no mucho.
- DENY no existe, ahora sería DROP.
- MASQ y REDIRECT no existen como destinos de paquetes.
- REJECT extendidos con más opciones
- LOG con más opciones, muy útil para monitorear y depurar
- ... y más que se pueden ver en el Howto y en otras páginas.

Elementos básicos

-Ordenes básicas:

iptables -F : efectivamente, flush de reglas
iptables -L : si, listado de reglas que se están aplicando
iptables -A : append, añadir regla
iptables -D : borrar una reglas, etc...

-Ejemplo de regla:

```
#Regla que acepta conexiones al puerto 80
iptables -A INPUT -i eth0 -s 0.0.0.0/0 -p TCP —dport www -j ACCEPT
```

ANATOMÍA DE LA REGLA:

iptables: commando iptables (no hay que olvidar que las reglas son un shell script)
-A: append, opción para añadir la regla
INPUT: estado del paquete (al entrar es input).

-i eth0: interfaz de red eth0
-s 0.0.0.0/0: dirección de acceso (cualquiera en este caso)
-p TCP: tipo de puerto
--dport: puerto de destino
-j ACCEPT: destino del paquete (se acepta, podría ser DROP, LOG, REJECT,..)

-Guía rápida de flags:
-s : source address. Ej: -s 192.168.1.0/24
-d : destino. Ej: -d 84.56.73.3
-p : tipo de protocolo(TCP,UDP,ICMP). Ej: -p TCP
--sport : puerto de origen

--dport: puerto de destino
-i = -in-interfase : el interfaz por el que se entra (eth0,eth1, ppp0,...)
-o = --out-interfase: el interfaz por el que se sale (eth0,eth1, ppp0,...)
-Notas:
-i se usa con reglas INPUT y FORWARD
-o se usa con reglas FORWARD y OUTPUT

A partir de estas normas básicas, conociendo la anatomía básica de una regla, y viendo ejemplos ya tenemos suficiente material para hacernos con el dominio de IPTABLES.

Ejemplos de Configuración

Ejemplo de Firewall Simple

Ahí va volcado:

```
#!/bin/sh
```

```
#####################################################################
## SCRIPT de IPTABLES                                              ##
## Pello Xabier Altadill Izura I+D+I+I en tiempo record            ##
## Investigación, Desarrollo, Innovación e IMPLANTACIÓN            ##
## Este script es de ejemplo y no es el mejor ejemplo,             ##
## pero funciona en RedHat 7.2 y es muy pedagógico                 ##
#####################################################################

## Notas para usuarios de IPCHAINS:
 # ipchains e iptables son módulos del kernel que
 # NO pueden convivir juntos
 # DENY ahora es DROP
 # Los LOG se guardan de otra forma
 echo -n Aplicando Reglas de Firewall...

## Instalando módulos
 modprobe ip_tables
 modprobe ip_nat_ftp
 modprobe ip_conntrack_ftp

## Variables
 EXTIF=”eth0” # La que va al router
 INTIF=”eth1” # La que va a la LAN

## Primeras reglas
 /sbin/iptables -P INPUT ACCEPT # INPUT se acepta por defecto MAL HECHO
 /sbin/iptables -F INPUT
 /sbin/iptables -P OUTPUT ACCEPT # OUTPUT se acepta por defecto, weno..
 /sbin/iptables -F OUTPUT
 /sbin/iptables -P FORWARD ACCEPT # FORWARD se acepta por defecto buf
 /sbin/iptables -F FORWARD
 /sbin/iptables -t nat -F

 ## se deniega 80 y se guarda log (ejemplo)
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport www -j LOG —log-prefix “IPTablesFW> “
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport www -j DROP

## Acceso al 3128 (proxy squid) desde LAN
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 3128 -j ACCEPT
 # El resto se tira
 /sbin/iptables -A INPUT -i $INTIF -s 0.0.0.0/0 -p TCP —dport 3128 -j DROP

 ## Acceso al 143 desde LAN
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 143 -j ACCEPT

## Acceso al ssh desde la LAN
 /sbin/iptables -A INPUT -i $EXTIF -s 213.195.64.0/24 -p TCP —dport 22 -j ACCEPT

## Acceso al ssh un rango externo
 /sbin/iptables -A INPUT -i $EXTIF -s 213.195.64.0/24 -p TCP —dport 22 -j ACCEPT
 # el resto se tira
 /sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p TCP —dport 22 -j DROP
## Acceso al puerto 25
 /sbin/iptables -A INPUT -i $EXTIF -s 213.191.89.0/24 -p TCP —dport 25 -j ACCEPT
 /sbin/iptables -A INPUT -i $INTIF -s 192.168.1.0/24 -p TCP —dport 25 -j ACCEPT
 /sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p TCP —dport 25 -j DROP
```

```
## FORWARD
 # Que me haga log de todo el forward
 /sbin/iptables -A FORWARD -j LOG

## He aqui el forward para la LAN, una regla mágica
 iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
 # Ese peazo de bit que hay que habilitar
 echo 1 > /proc/sys/net/ipv4/ip_forward
```

Ejemplo de Firewall más Completo

Este ejemplo es algo más serio, ya que la regla de input por defecto es DROP. Esta política de reglas es la más segura, ya que por defecto se denegará TODO, y poco a poco se van habilitando las entradas precisas.

```
#!/bin/sh
## SCRIPT de IPTABLES
## Pello Xabier Altadill Izura
echo -n Aplicando Reglas de Firewall...

## Paramos el ipchains y quitamos el módulo
 /etc/rc.d/init.d/firewall stop
 rmmod ipchains

## Instalando módulos
 modprobe ip_tables
 modprobe ip_nat_ftp
 modprobe ip_conntrack_ftp

 ## Variables
 IPTABLES=iptables
 EXTIF=”eth1”
 INTIF=”eth0”

## En este caso,
 ## la tarjeta eth1 es la que va al ROUTER y la eth0 la de la LAN

## Primeras reglas
 /sbin/iptables -P INPUT DROP
 /sbin/iptables -F INPUT
 /sbin/iptables -P OUTPUT ACCEPT
 /sbin/iptables -F OUTPUT
 /sbin/iptables -P FORWARD ACCEPT
 /sbin/iptables -F FORWARD
 /sbin/iptables -t nat -F

### En principio, si las reglas INPUT por defecto hacen DROP, no haria falta
 ### meter más reglas, pero si temporalmente se pasa a ACCEPT no esta de más.

## Todo lo que viene de cierta IP se deja pasar (administradores remotos...)
 /sbin/iptables -A INPUT -i $EXTIF -s 203.175.34.0/24 -d 0.0.0.0/0 -j ACCEPT

## El localhost se deja
 /sbin/iptables -A INPUT -i lo -j ACCEPT
 /sbin/iptables -A OUTPUT -o lo -j ACCEPT

## Aceptar al exterior al 80 y al 443

# Permitir salida al 80
```

```
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 80 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 80 -j ACCEPT
# Permitir salida al 443
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 443 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 443 -j ACCEPT

## SALIDA SMTP - Para que el servidor se pueda conectar a otros MTA
# Permitir salida SMTP
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 25 -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 25 -j ACCEPT

## SALIDA FTP - Para que el servidor se pueda conectar a FTPs
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 21 -m state —state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 21 -m state —state NEW,ESTABLISHED -j ACCEPT
# ftp activo
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 20 -m state —state ESTABLISHED,RELATED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —dport 20 -m state —state ESTABLISHED -j ACCEPT
# ftp pasivo
/sbin/iptables -A INPUT -i $EXTIF -p tcp —sport 1024:65535 —dport 1024:65535 -m state —state ESTABLISHED -j ACCEPT
/sbin/iptables -A OUTPUT -o $EXTIF -p tcp —sport 1024:65535 —dport 1024:65535 -m state —state ESTABLISHED,RELATED -j ACCEPT

## El acceso al 19720 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 19720 -j DROP

## El acceso al 19720 desde dentro, ACEPTADO
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 19720 -j ACCEPT

## El acceso al 19721 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 19721 -j DROP

## El acceso al SSH desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 1972 -j DROP

## El acceso al SMTP desde dentro, permitido.
/sbin/iptables -A INPUT -i $INTIF -p tcp —dport 25 -j ACCEPT
## El acceso al SMTP desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -d 0.0.0.0/0 -p tcp —dport 25 -j DROP

## Acceso al 80 desde el interior ACCEPTADO PARA DOS IPs
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.11/32 -p tcp —dport 80 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.54/32 -p tcp —dport 80 -j ACCEPT
## Acceso al 80 desde el interior DENEGADO PARA EL RESTO
#/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 80 -j DROP

## Acceso al PROXY
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 8082 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 8082 -j ACCEPT
/sbin/iptables -A INPUT -s 127.0.0.0/8 -p tcp —dport 8082 -j ACCEPT

# Desde el exterior denegado
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 8082 -j DROP

## Acceso a POP3 e IMAP desde el EXTERIOR, DENEGADO
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 110 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.9.0/24 -p tcp —dport 143 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 110 -j ACCEPT
/sbin/iptables -A INPUT -i $INTIF -s 192.168.10.0/24 -p tcp —dport 143 -j ACCEPT
```

```
## Acceso a POP3 e IMAP desde el EXTERIOR, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 110 -j DROP
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 143 -j DROP
## Acceso al 8082 desde fuera, DENEGADO
/sbin/iptables -A INPUT -i $EXTIF -s 0.0.0.0/0 -p tcp —dport 8082 -j DROP

## FORWARD
# Que me haga log de todo el forward
#/sbin/iptables -A FORWARD -j LOG

# He aqui el forward
## Norma general
##iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.11/32 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.16 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.54/32 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.9.0/24 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.10.0/24 -j MASQUERADE

# Habilitar el forwarding
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Reglas de Protección

Tranquilos , aquí no vamos a hablar de la última maza de cartas Magic que me he comprado, ni de formulas para salir airosos jugando a La Llamada de Cthulhu, entre otras cosas porque no juego a esas cosas raras. En esta breve sección listamos algunas reglas para proteger nuestro equipo y por extensión la red, de ciertos ataques muy habituales en las redes como el smurf y otras formas de inundación y DoS.

¿Es recomendable usar todas estas normas? Según como administremos el nivel de paranoia.

Nota: hay que dar valores concretos a las variables $

```
# Deshabilitar broadcast
/bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Deshabilitar el ping... quizá discutible.
/bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all

# Deshabilitar la redirección del ping
/bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects

# Registrar los accesos extraños, paquetes falseados, etc..
/bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
# Anti-flooding o inundación de tramas SYN.
iptables -N syn-flood
iptables -A INPUT -i $IFACE -p tcp —syn -j syn-flood
iptables -A syn-flood -m limit —limit 1/s —limit-burst 4 -j RETURN
iptables -A syn-flood -j DROP
# Guardar los accesos con paquetes fragmentados, recurso utilizado para tirar
# servidores y otras maldades (bug en Apache por ejemplo)
iptables -A INPUT -i $IFACE -f -j LOG —log-prefix "Fragmento! "
iptables -A INPUT -i $IFACE -f -j DROP

# Anti-spoofing (falseo de ip origen)
```

```
iptables -A INPUT  -i $IFACE -s $IPADDR -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_A -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_B -j DROP
iptables -A INPUT  -i $IFACE -s $CLASS_C -j DROP
iptables -A INPUT -i $IFACE -s $CLASS_D_MULTICAST -j DROP
iptables -A INPUT -i $IFACE -s $CLASS_E_RESERVED_NET -j DROP
iptables -A INPUT  -i $IFACE -d $LOOPBACK -j DROP
iptables -A INPUT -i $IFACE -d $BROADCAST -j DROP
(bien explicado en el script de linuxguruz)
# Proxy Transparent: peticiones al puerto 80 redirigir al SQUID(3128)
iptables -t nat -A PREROUTING -p tcp -s 0.0.0.0/0 —dport 80 -j REDIRECT —to 3128
```

Servidor de Proxy SOCKS

SOCKS le permite a los usuarios acceso al Internet desde atrás de un firewall. Derivada de la palabra sockets, SOCKS centraliza el control de la red sobre la actividad del Internet, permitiendo acceso limitado a computadoras específicas y a través de autentificación (login) a actividades en particular en el Internet. SOCKS es una herramienta más de administración y protección en una red.

Hay dos partes a los SOCKS: el cliente y el servidor. El programa del lado del cliente se ejecuta en el computador detrás del firewall y se conecta al servidor de SOCKS para hacer particiones de acceso al Internet. El servidor de SOCKS se coloca entre la red detrás del firewall y el Internet. Un cliente contacta el servidor, el cual maneja la petición y transfiere la información entre el cliente y el sitio de Internet.

Cuando un cliente hace una petición para obtener acceso al Internet, este envía la siguiente información:
- Comando de petición de conexión
- Número de Puerto
- Dirección de IP Requerida
- Nombre de Usuario del sistema requiriente

Ya en este punto, la administración de la red entra en una etapa importante. La información recibida desde el cliente es comparada a la de las listas de control, creadas por el administrador. Si el cliente posee privilegios de acceso al Internet, la petición es servida y el servidor de pasa la información al sitio de Internet. La data es enviada de regreso desde el sitio de Internet arriva al servidor SOCKS. El servidor pasa la data al cliente después de revisar los permisos del cliente nuevamente.

SOCKS es gratis en el Internet y puede ser descargado desde el sitio Web http://www.socks.nec.com.

Cada cliente que usa SOCKS debe tener su propio archivo de configuración. Al momento de hacer una petición, el archivo /etc/socks.conf es revisado línea por línea. Una vez la conexión requerida es encontrada, la acción especificada en el archivo es efectuada. El archivo contiene los siguientes comandos:

deny

Este específica que la dirección TCP/IP que iguala esta línea debe ser permitida. Pero, usted no debe usar este operador en vez de protección apropiada de firewall. Aunque la dirección está bloqueada de acceso, el cliente puede usar un programa no vinculado a la librería de SOCKS, sobrepasando la seguridad implementada en el servidor SOCKS.

direct

Este pasa la dirección TCP/IP especificada a través del firewall sin aplicar ningún filtro.

sockd

Este instruye a la librería SOCKS para que la conexión TCP/IP sea enviada al daemon sockd donde, después de revisar, este presumiblemente envía la petición fuera de la red.

El archivo /etc/sockd.conf controla cuales conexiones TCP/IP son pasadas al daemon sockd y eventualmente fuera del firewall. El daemon sockd normalmente se ejecuta en el computador situado entre la red interna y la red externa, por lo general en la máquina que actúa como el firewall.

El archivo /etc/sockd.conf es un archivo de texto plano. Cada línea contiene hasta 1,023 caracteres y utiliza el siguiente formato:

[allow / deny] [?=auth] [*=username | filename]
source-address source-mask
[destination-address destination-mask]
(.operator destination-port]
[:shelI -command]

allow/deny
Esta opción específica si las conexiones en esta línea son permitidas o no.

?=auth
La política de identificación especificada al inicio del programa puede ser obviadas con esta opción. Usando ?=I causa que la conexión sea rechazada si el cliente esta usando ident o si el nombre retornado por ident no es el mismo que el que es enviado por el cliente SOCKS. Un ?=i causa que sockd rechace la conexión si el nombre retornado por ident es diferente al nombre enviado por el cliente SOCKS.

***=username | filename**
Esta opción causa que la regla iguale a un nombre de usuario en particular.

source-address source-mask
Esta opción permite a igualar una dirección IP única o a un grupo de direcciones IP. la dirección de origen puede ser en la anotación de cuatro octetos separados por puntos (100.99.98.97) o el nombre del host. La mascara de origen debe ser en la anotación separada por puntos.

operator destination-port
Un puerto en particular o un rango de puertos TCP/IP puede ser especificado con estas opciones. Por ejemplo, "eq 20" establece una conexión igual al puerto de data del ftp. La siguiente es una lista de los operadores disponibles:

eq	Igual a
neq	No es Igual a
lt	Menor que
le	Menor que o Igual a
gt	Mayor que
ge	Mayor que o Igual a

:shell-command
Esta opción específica cual comando usar cuando la línea es igualada. El comando es ejecutado por el shell bash (/bin/sh).

route y traceroute

El sistemas GNU/Linux, la configuración de enrutamiento estático se logra a través del uso del comando route. Este permite la modificación o visualización de la tabla de enrutamiento. Esta también puede ser usada para ayudar a diagnosticar problemas de enrutamiento. Otro comando muy útil que puede ser usado para diagnosticar problemas de enrutamiento y conectividad es el comando traceroute.

Comando route

El comando route de GNU/Linux es por lo general usado para designar una dirección IP estática a una red o computador específica usando una interfase de red. El comando route puede ser usado en conjunto con un sistema que recibe su dirección IP desde un servidor DHCP. El sintaxis del comando route es route [opción] [argumento] [dispositivo]. Aquí le presentamos algunas de las opciones más comunes usadas con el comando route. Para una lista más completa refierase a la páginas man del comando route.

-n	**Despliega los hosts por dirección IP (no se muestran los nombres de host)**
add	**Agrega una a la tabla de enrutamiento**
del	**Elimina una ruta en la tabla de enrutamiento**
net	**Conecta la ruta a una red**
host	**Conecta la ruta a un máquina host**
gw	**Específica por cual gateway enrutar en el host o en el sistema de red**

Este es un ejemplo de algunas de estas opciones:

route add -host 192.168.1.0 eth0

En este ejemplo conectamos a otro host en la red 192.168.1.0 usando la primera interfase de Ethernet.

Al escribir el comando route sin ningunas opciones nos muestra la configuración de la tabla de enrutamiento actual. Por ejemplo:

route

The following is the Kernel IP routing table:

Destination	Gateway	Genmask	Flags	Ref	Use	Iface
192.168.1.0	0.0.0.0	0.0.0.0	U	0	0	eth0
default	gw.serv.com	0	UG	0	0	eth0

La primera y segunda columnas muestran las direcciones IP o los nombres de de el host/red y gateway, respectivamente. La columna Genmask nos da la mascara de la subred para la ruta. La columna Flags tiene tres opciones: U (la conexión esta activa), H (el destino es un host), y G (el destino es un gateway). La columna Metric muestra la distancia al destino, el cual es usado por algunos daemons enrutadores para calcular la mejor ruta. La columna Ref es usada por UNiX para indicar los números de referencia del router. La columna Use muestra cuantas veces el kernel busco la información de enrutamiento. Por último, la columna Iface muestra cual dispositivo de red es usado para mandar los paquetes a esa dirección o ruta.

Comando traceroute

El comando traceroute es una herramienta invaluable para determinar donde está ocurriendo un problema en la red, traceroute envía un paquete al destino especificado y reporta todos los saltos que se dan en la ruta. Un salto puede no responder por varias razones, incluyendo exceso de tráfico o que físicamente el sitio Web este abajo.

El traceroute utiliza el campo TTL (Time to Live) en el encabezado del paquete para determinar la información de la tabla de enrutamiento. El campo TTL limita a cuantas computadoras o gateways un paquete puede proceder antes de ser descartado. Cada vez que un enrutador reenvía un paquete, esta disminuye el TTL por 1. Si un paquete llegase a cero, el paquete será descartado (dropped) y el router envía un mensaje ICMP haciendole saber al host que fué descartado.

El comando se inicia enviando un paquete con un TTL de 1. Si el primer gateway o host no es el destino asignado, el gateway disminuye el TTL por 1 y envía una respuesta a traceroute. El traceroute da salida a la dirección de la primera repuesta y crea otro paquete. Al nuevo paquete se le da un TTL de 2 y es enviado. El primer gateway pasa el paquete más adelante, reduciendo el TTL por 1. En la próxima pasarela, el TTL, es ahora 1, a este punto, la pasarela reduce el TTL, a 0 y retorna una respuesta al router, traceroute incrementa el TTL, desplegando la repuesta desde los gateways intermedio hasta que el paquete arrive a su destino final. Por defecto, el número de saltos es 30 pero puede ser cambiado usando la opción -m (max_ttl).

Las siguientes opciones son comunes de tracecoute:

-f Establece el valor inicial TTL, permitiendole saltar enrutadores cerca del host
-m Establece el máximo número de saltos
-n No efectúa Búsqueda de nombre; sino, sólo da salida a direcciones numéricas de IP

El comando tracetoute produce salida en un formato similar al del comando ping:

```
# traceroute  abiertos.org
traceroute to abiertos.org (206.123.67.148), 30 hops max, 38 byte packets
 1  adsl-200-1.tricom.net (200.42.200.1)  1451.603 ms  210.628 ms  90.021 ms
 2  Leased-Lines-208-93.tricom.net (200.42.208.93)  9.581 ms  11.010 ms  10.798 ms
 3  Leased-Lines-208-174.tricom.net (200.42.208.174)  13.945 ms  91.611 ms  11.987 ms
 4  sl-gw11-orl-4-0-TS3.sprintlink.net (160.81.34.65)  139.749 ms  1440.340 ms  1478.104 ms
 5  sl-bb20-orl-0-0.sprintlink.net (144.232.2.232)  1423.170 ms  1431.891 ms  1431.974 ms
 6  sl-bb21-atl-10-3.sprintlink.net (144.232.19.73)  1647.392 ms  1576.853 ms  605.996 ms
```

7 sl-gw25-atl-15-0.sprintlink.net (144.232.22.14) 126.988 ms 172.011 ms 114.977 ms
8 sl-timewarner-9-0.sprintlink.net (144.223.62.62) 61.550 ms 1351.235 ms 1546.623 ms
9 core-01-ge-1-2-0-1.atln.twtelecom.net (66.192.243.40) 1612.145 ms 1522.711 ms 1445.372 ms
10 core-02-so-4-0-0-0.dlfw.twtelecom.net (66.192.255.24) 1606.380 ms 1453.472 ms 1585.293 ms
11 hagg-02-ge-3-3-0-504.dlfw.twtelecom.net (66.192.253.125) 1466.255 ms 1460.830 ms 1525.948 ms
12 66-195-98-102.gen.twtelecom.net (66.195.98.102) 1565.522 ms 156.115 ms 93.964 ms
13 206.123.64.46 (206.123.64.46) 221.905 ms 1633.271 ms 1568.761 ms
14 abiertos.org (206.123.67.148) 467.253 ms 82.485 ms 209.568 ms

La primera parte de información es denotar el número de saltos (hops) para llegar desde el origen al router localizado en la dirección dada como argumento. El número de saltos es seguido por el nombre de del router. Si el nombre del router no esta disponible, la dirección IP es desplegada. Luego es la dirección IP del gateway. Los próximos tres campos muestran el tiempo de respuestas, o el tiempo que el paquete toma para hacer el todo el viaje por la red de ida y vuelta desde su sistema y de vuelta.

Modo Promiscuo

El Ethernet usa un protocolo llamado CSMA/CD (Carrier Sense Multiple Access/Collision Detection (CSMA/CD). El termino Carrier Sense se refiere al proceso por el cual una estación revisa a ver si otra estación esta usando la línea antes de ella enviar. Si otra estación se encuentra ejecutando antes de que la transferencia de data inicie. Si no se detecta data, la estación empieza a transmitir. Si hay tráfico, la estación espera por el próximo tiempo de línea reposo. Acceso multiple significa que cada estación esta conectada a un único cable o conjunto de cables que forman una única ruta. El termino Collision Detection se refiere a la manera por la cual los temas de colisiones son resueltas. Aunque las estaciones revisan por tráfico antes de empezar a transmitir, las estaciones pueden detectar que no hay tráfico y empiezan a enviar simultáneamente. Una vez ocurre una collisión y cada estación se percata de esto (embedido en la data es un medio para detectar si la data llego a su destino que causa que el destinatario responda en un tiempo en específico), ambas estaciones dejan de transmitir y esperan un lapso de tiempo al azar antes de tratar de retransmitir.

Ethernet esta basada en la arquitectura bus. así que, los cables de Ethernet lleva tráfico de data que esta destinado para multiple computadoras. Cada computador escucha y acepta data direccionada específicamente a ella por su dirección MAC. Las computadoras por lo general ignoran paquetes que no estén direccionados a ellos. Pero, un sistema puede ser configurado para escuchar a toda la data que pasa sobre el cable o la data dirigida a otros sistemas. Un sistema esta en modo promiscuo cuando este lee data dirigida a otros sistemas. Existen razones legitimas para que un sistema sea establecido en el modo promiscuo, como puede ser prueba de cables para detectar deficiencias de conexiones análisis de tráfico (tipo de tráfico, los puertos usados , volumen, entre otros.).

Ejercicio 3-2: Conceptos de Enrutamiento

En este ejercicio, trabajaremos con una tabla de enrutamiento para determinar como los paquetes serán enrutados entre las redes. No se proveen soluciones a es ejercicio.

1. Considere la siguientes entradas en la tabla de enrutamiento.

Route	**Netmask**	**Destination**
132.10.56.4	**255.255.255.255**	**142.12.201.2**
132.10.0.0	**255.255.0.0**	**142.12.201.1**
132.10.20.0	**255.255.255.255**	**132.10.4.18**
15.0.0.0	**255.0.0.0**	**132.1 0.4.1 B**
132.10.56.5	**255.255.255.255**	**132.10.56.4**
132.10.20.5	**255.255.255.255**	**142.12.201.2**
142.12.0.0	**255.255.0.0**	**Local interfase**
0.0.0.0	**0.0.0.0**	**142.12.201.2**

Donde (por ejemplo, a cual dirección IP) el enrutador entregará los paquetes direccionados a las siguientes destinaciones? (Use la tabla anterior para responder.)

15.127.243.8
132.10.56.4
132.10.20.3
132.10.20.5
132.10.200.3
37.92.129.1

2.- Considere el siguiente diagrama de una LAN.

Una máquina de prueba (A) es colocada en la LAN y se desea revisar su configuración IP. Se logra esto efectuando el comando ping a la máquina (B) en la misma red. El echo retorna; todo es bien entonces. Entonces intentamos hacerle ping ping a la máquina (C) que se encuentra en otra red. Esta vez el ping falla- ningunos son retornados. Verificamos que mi máquina tiene una ruta desde A a C via el router. ¿Entonces por qué el ping no funciona?

3. En la Compañía X, es hora de decidir en el enrutamiento entre el enrutamiento de la redes de X.

La tarea es dibujar las tablas de enrutamiento para los routers en el diseño de la red de su grupo. Si usted aún no ha asignado algunas clases de IP a la redes en su plan, deberá hacerlo ahora.

¿Percibe usted alguna área de problema /temática?

Ejercicio 3-3: Examinar las Tablas de Enrutamiento

No se proveen soluciones a este ejercicio. Podemos examinar las tablas de enrutamiento del sistemas usando el comando netstat con la opciones -r o -e:

$ netstat -r   # route -e

A menudo es útil desplegar los hosts y las redes en la tabla en termino de sus direcciones IP y no sus nombres. Este es muy útil en particular cuando estamos usando servidores de nombre de dominio como es el DNS (lo cual se discutirá más adelante). Para efectuar esta acción, escriba el siguiente comando:

$ netstat -rn

Verá una entrada para la interfase de red y también una para la interfase loopback. En esta etapa, podrás ver la pequeña diferencia entre las dos salidas.

Pruebe las opciones -v y -ee del comando route; ¿qué información nos arroja cada una?

Sin una entrada en la tabla de enrutamiento que nos diga como llegar a una red o a un host, no nos podremos comunicar no con la red ni el host. Para ilustrar esto, escriba el siguiente comando:

$ ping Dirección_IP

Donde la Dirección_IP es la dirección de uno de los sistemas en otro segmento de la red. Note los errores de mensajes que se despliegan. Ahora ejecute este comando y también observe los mensajes de errores:

$ telnet Dirección_IP

PROTOCOLOS DE ENRUTAMIENTO DINÁMICO

Para comprender las teorías y el funcionamiento de enrutar y los routers debe entender como es que ellos actualizan y mantienen sus tablas de enrutamiento, que es lo que le permite a ellos poder dinámicamente enrutar los paquetes.

En esta sección discutiremos los siguientes tópicos:

- ¿Por qué Enrutar dinámicamente?
- Enrutamiento Dinámico: El Concepto
- Protocolos de Enrutaramiento

- El ICMP y la Redirección de los Mensajes
- La Clasificación de los Protocolos
- Concepto de un Sistema Autónomo
- Protocolos de Interior y Exterior
- Implementaciones UNiX

¿Por qué Enrutar Dinámicamente?

Para redes pequeñas, el enrutamiento estático es por lo general suficiente. Muchas redes contienen uno o dos routers . Estas redes pueden ser administradas normalmente usando enrutamiento estático; esto es, las tablas de enrutamiento son entradas a mano y mantenidas manualmente.

A medida que las redes crecen, así incrementa las posibilidad de riesgo. Consideremos una red con cientas de computadoras y docenas de routers; se convierte en casi imposible el mantenimiento y la consistencia de las tablas de enrutamiento.

El propósito de usar enrutamiento Dinámico es para que las redes se actualicen automáticamente. Las redes cambian: nodos se agregan y se eliminan, se crean y eliminan enlaces, a veces temporalmente. Ya que las redes son una colección de computadoras (routers), entonces ¿Por qué es que estas computadoras no puede monitorear y actualizarces para mantener un enrutamiento sincronizado?

Enrutamiento Dinámico: El Concepto

En redes a gran escala, la confianza se coloca sobre los sistemas de gateways para que esto mantengan actualizada la información de las rutas que están en todo el Internet. Esta información es potencialmente volátil ya que los enlaces frecuentemente se colocan dentro y fuera de servicio, especialmente en ambiente de redes la área amplia.Y por estos no es práctico, mantener las tablas de enrutamiento de los gateways manualmente.

Cada enrutador propaga la información de routing (o disponibilidad) a sus vecinos. Este también actualiza su tabla de enrutamiento de la información presentada a el por los otros routers. Más a menudo, los gateways implementan lo que es conocido como enrutamiento dinámico para mantener sus tablas correctas. Esto se logra haciendo que los sistemas de gateway ejecuten un proceso especial, conocido como enrutamiento de servidor (routing server) o daemon de enrutamiento (routing daemon).

El trabajo de este proceso es intercambiar información que puede ser usada para determinar las rutas a otras redes. así también como comunicarse con daemons de enrutamiento de otros sistemas, el proceso local también es responsable por el mantenimiento de las tablas de enrutamiento en el sistema local basado en la información que este recibe de los otros sistemas.

De esta manera es mucho más simple mantener las tablas de enrutamiento en un sistema al día con la configuración de la red de Internet.

El Proceso de Enrutamiento Dinámico

Es importante entender que enrutamiento Dinámico no cambia la manera en que la capa de Redes efectúa su función de enrutar paquetes, el mecanismo de enrutamiento. Enrutamiento Dinámico determina la información colocada en la tabla de enrutamiento. Si se agrega una nueva red esta es agregada o si un enlace se cae, la tabla de enrutamiento será modificada debidamente. Si se encuentran multiple rutas al mismo destino, la mejor ruta es elegida e insertada en la tabla de enrutamiento. Esta elección se lleva acabo de acuerdo con las políticas de enrutamiento. Cada enrutador contiene un proceso, llamado daemon de enrutamiento,

que efectúa estos cambios (este ejecuta las políticas de enrutamiento).

La información de enrutamiento es propagada de enrutador a enrutador en mensajes de actualización como parte del protocolo de enrutamiento.

Protocolos de Enrutamiento

Los protocolos se comunican usando protocolos de enrutamiento. Estos han evolucionado a través de los años. Ahora los podemos clasificar en dos grupos como protocolos de estado de enlace (link-state protocols) y protocolos de vector de distancia (vector-distance). Hay algunos requerimientos principales que deben por lo general ser satisfecho.

Convergencia

Esto es para cuando las topología de red cambian. Las tablas de enrutamiento necesitan ser actualizadas para reflejar esos cambios en una forma ordenada. El proceso de actualizar las tablas de enrutamiento cuando las redes cambian debe suceder suficientemente a menudo para que las tablas de enrutamiento reflejen el verdadero estado de la red; con esto se quiere dejar dicho que, las tablas de redes deben converger en el estado de la red rápidamente.

Consistencia

Las inconsistencias entre los routers pueden potencialmente convertir una destinación inalcanzable. Un ejemplo es: un router anuncia una ruta que ha sido eliminada desde un router más profundo en la red.

Seguridad (Autenticación)

Como los protocolos de enrutar actualizan las tablas de enrutamiento, existen un potencial desafortunado para un atacante desde otra red para enviar una actualización de enrutado a un router. El efecto es que los paquetes no son ya enrutados o pueden ser enrutados al destino equivocado.

ICMP y la Redirección de Mensajes

La forma más simple de enrutamiento Dinámico es proveída por el mecanismo de redirección de ICMP. Bajo circunstancias normales, sistemas hosts son inicializados con información minina de enrutamiento. Como descripto anteriormente, un host normalmente sabrá como enrutar paquetes para la red local y también la dirección de un sistema de gateway único al cual todos los paquetes deberán ser enviados.

En algunos casos, este talvez no es la mejor manera de enrutar paquetes. En el ejemplo anterior, al host H1 se le ha dicho que todos los paquetes para otras redes deben ser enrutado a través del gateway G1. Pero, puede ser que la ruta desde Gl a H2 vía G2 es costosa en tiempo o recursos. En este caso, una mejor ruta sería enviar los paquetes por G3, de donde esta puede ser enviada directamente al host H2.

Si la línea entre G1 y G2 está lenta, las tablas de enrutamiento en G1 serán configurada para enviar paquetes para H2 vía G3. Cuando G1 recibe el paquete desde H1 y entonces decide enrutarlo vía G3, este nota que H1 y G3 son las misma red. así que, para evadir este extra salto para paquetes futuro, G1 puede enviar un mensaje especial ICMP de regreso a H1 diciendole que use G3 como un gateway para arribar a H2. Este mensaje es conocido como un mensaje de redirección. Al recibir el mensaje, H1 actualiza su tabla de enrutar para incluir detalles de la nueva ruta.

Use esta técnica, es posible inicializar un host con información minina de enrutamiento, sabiendo que este aprenderá a cerca mejores rutas desde los gateways con el pasar del tiempo.

Clasificación de Protocolos

Los protocolos de enrutamiento han evolucionado a través de los años. Observaremos mayormente a dos

tipos:
• Protocolos Vector-Distance
• Protocolos Link-State

Los protocolos difieren en dos maneras. Primero, ellos difieren en el tipo de información que ellos comunican entre los routers. Segundo, ellos difieren en la manera que ellos usan esta información para actualizar las tablas de enrutamiento.

Protocolos Vector-Distance (Vector-Distancia)

Los protocolos anteriores vector-distance han sido en muchos casos han sido remplazados por los protocolos link-state. Ejemplos son RIP y Hello. Los routers propagan la información de destinación/cuenta de saltos a los otros routers.

Los protocolos vector-distance esencialmente reenvían (forward/inundación de red) , rutas a través de la red de un router a otro router. Las rutas son enviadas como una tabla, de la cual la primera columna es el destino final (vector) y de cual la segunda columna es la metric (distance). Si el destino no existe en la actualización de la tabla de enrutamiento, una nueva ruta será agregada. Si la lista de enrutamiento actualizada contiene una ruta más corta, la en la tabla es reemplazada. Si la lista actualizada en más larga y el router actualmente enruta vía el router enviando la actualización de enrutamiento, entonces la ruta es reemplazada con la ruta más larga.

RIP es uno de los protocolos de enrutamiento vector-distance más popular. Esto no es debido a cualquier mérito técnico sino debido a que la implementación de RIP fué puesta en disposición a través de la versión de UNiX 4.2 BSD. Esta implementación fué copiada y distribuida ampliamente. De hecho, no hubo originalmente ninguna especificación de RIP. Un RFC fué creado años después, derivado de la implementaciones ya existente.

Para elegir entre rutas alternativas al mismo destino, un valor metric de cuenta de saltos (número de etapas en la ruta) es usado. Las diferencias en tiempo (por ejemplo, entre un enlace rápido y uno lento) puede ser incorporado a través de la escogencia del valor de metric. RIP considera que un valor de más de16 saltos es efectivamente inalcanzable, por esto limitando el tamaño de la red en las que RIP puede ser usado.

Existe una versión nueva de RIP, RIP-2, cual es descripta en el RFC 1388. Este presenta ciertas mejoras sobre el RIP original. En áreas tales como el manejo de las subredes de tamaños diferentes en el mismo host. Este también incorpora algunas nuevas características, así como el soporte para multicasting y una ficha o tag de ruta para hacer que opere mejor con otros protocolos de gateway exterior.

Protocolos de Link-State

Un perfecto ejemplo es el OSPF de Dijkstra, algoritmo de Búsqueda, de la red completa de la primera ruta abierta más corta (Open Shortest Path First). Formas más avanzadas de protocolos de enrutamiento usan el método de link-state. Con OSPF, actualizaciones no contienen rutas, solamente el estado de los enlaces directamente conectados.

Hay varias ventajas con enrutamiento link-state, que entre ellas se incluyen:

- Cada enrutador re-calcula sus rutas independientmente y no dependen de equipos externos.
- Enrutadores Link-state garantizan que convergen.
- Mensajes de estado de enlace pasan a través de la red sin ser cambiados, haciendolos así más fácil depurarlos.
- Enrutamiento de Link-state hereda su escabilidad del hecho que el tamaño de los mensajes no depende en el número de enlaces desde el enrutador.
- Enrutamiento de Link-state genera menos tráfico que el enrutamiento de distance-vector porque los men-

sajes son por lo general más pequeños.

Cuando el estado cambia, el router reconstruye su gráfica y vuelve a calcular las rutas usando el algoritmos OSPF de Dijkstra, el calcula la ruta más corta al destino final.

Otros Protocolos de Enrutamiento

Uno de los protocolos de interior más nuevo es el ICMP (Router Discovery Protocol). Este sistema provee un mecanismo para que un host elija el enrutador por defecto más apropiado. El protocolo trabaja en un mecanismo basado en anuncios y solicitudes. Cuando un host inicia un software TCP/IP, esta hará un broadcast a un router solicitandole un mensaje. Cualquier enrutador que recibe este mensaje devolverá un paquete anunciante que contiene su dirección IP y un valor preferencial. El host escuchante seleccionará el enrutador con el mejor valor preferencial e instalará ese en su tabla de enrutamiento como el router por defecto.

Además de responder a paquetes de solicitudes, los routers enviarán paquetes de broadcast cada 7-10 minutos. El descanso es al azar entre los envíos de estos paquetes para evitar colisiones con otros routers también anunciandose. Un router puede alterar su valor de preferencia para reflejar el estado de sus intefases. Los hosts escuchantes pueden entonces decidir si otro router es mejor y, entonces, proceder a alterar su entrada de router por defecto. Si un host no recibe un paquete desde su enrutador por defecto cada 30 minutos, eliminará (drop) esa ruta y usaría el próximo mejor router en su lista.

Otro protocolo de interior similar a RIP es el llamado Hello. Este protocolo usa una medida de espera de tiempo (time-delay) para elegir entre rutas alternativas y no una distancia. Este fué el protocolo original en la NSFNET (National Science Foundation Network). Es usado muy poco hoy en día.

Hay varios otros protocolos de enrutamiento que usted puede enfrentar. Ejemplos de estos serian los protocolos OSI, IDRP (Interdomain Routing Protocol) y ISIS (Intermediate System to Intermediate System).

Conceptos de Sistemas autónomos

Los sistemas autónomos pueden ser descriptos como una colección de redes y enrutadores debajo de un sólo control administrativo. Internets a gran escala son organizadas en base en una forma jerárquica, con una red central (núcleo) proveyendo un método para la interconexión de las demás redes que la constituyen. La administración de las subsecciones del Internet es a menudo delegada, así reinsertando la estructura jerárquica.

Desde el punto de vista del enrutamiento, es también deseable ver el Internet como una jerarquía de internets más pequeñas.

Cada uno de estos será administrado por una autoridad única independiente, por esto es que son conocidos como sistemas autónomos. Dentro de un sistema autónomo, pueden existir varios otros sistemas más pequeños enlazados por gateways. Estos gateways necesitan saber las rutas entre estas redes en el sistema autónomos pero no necesariamente de los otros sistemas autónomos. Estos son conocidos como gateways de interior.

Cada uno de los sistemas autónomos esta enlazado a la red central “core” por un gateway, cual debe saber por lo menos una información resumida de los otros sistemas autónomos. Estos son conocidos como gateways de exterior.

Para enviar un paquete a un sistema en otro sistema autónomo, un sistema de origen utiliza las tablas de enrutamiento en los gateways de interior de su propio sistema autónomos para llegar hasta en gateway de

exterior. Desde aquí el paquete puede ser reenviado al apropiado gateway de exterior para ser destinado al sistema autónomo donde el gateway de interior puede dirigirlo a su correcta destinación.

Diferentes protocolos para el intercambio de información de enrutamiento entre los sistemas autónomos han sido desarrollado, con uso dependiendo si el gateway en uso es interno o externo. Las características principales de estos protocolos incluyen:
- Libertad de uso de cualquier arquitectura de enrutamiento interno
- Ocultar la estructura interna
- Uno o más enrutadores intercambiando información sobre el alcance de sistemas autónomos a otros sistemas autónomos.

Gateways Core

Tradicionalmente los sitios eran conectados a un backbone de Internet. En los inicios del Internet moderno, una red backbone singular central interconectaba todos los sistemas autónomos juntos. Esto significaba que el tráfico entre los sistemas tenían que compartir el único backbone.

La administración del backbone era centralizada y por esto, relativamente fácil de controlar. Pero, este modelo rápidamente confrontó los problemas de escabilidad con todo el tráfico compartiendo un único backbone. Para aliviar el problema de ancho de banda, se agregaron más backbones.

Para mantener el tráfico fluyendo apropiadamente, el intercambio a gran escala de data enrutada tenia que tomar lugar entre los gateways core gateways en el backbone. A medida que nuevas redes fueron agregandose rápidamente, las tablas de enrutamiento crecieron a tamaños inmanejable.

Peer Backbones

La solución, la cual aún es la estructura básica del Internet de hoy, fué convertir este modelo de único backbone en una federación de arbitrariamente interconectados sistemas autónomos de tamaños diferentes. Esto simplifica las necesidades de enrutamiento (cada gateway solamente necesita saber acerca de sus vecinos, no todas las rutas posibles) y drásticamente reduce la demanda sobre cualquier de los backbones en particular.

Protocolos de Interior y Exterior

Los protocolos de enrutamiento de Internet pueden ser clasificado como interior o exterior, dependiendo si el enrutamiento es dentro o entre las organizaciones.

Protocolos Internos:
- Construir un imagen completa de su sistema autónomo.
- Usar rutas por defecto para tráfico no local.
- Trabajar a un nivel técnico.

Protocolos Externos:
- Trabajar con la información de alcance resumiendo las rutas a todo el sistema en el Internet.
- Puede que tenga que tomar decisiones de enrutamiento por razones no técnicas (otras que no sean de conectividad); Por ejemplo, una corporación con múltiples gateways al Internet probablemente no se desea que sean usados para el tráfico normal.

Protocolos de gateway Exterior permiten que gateways en las fronteras de sistemas autónomos intercambian información de las rutas dentro de los sistemas. Esta información entonces puede ser incorporada den-

tro de las decisiones de enrutamiento tomadas para transferir paquetes de un sistema autónomo a otro.

El protocolo original de ARPANET para esto era el EGP (Exterior Gateway Protocol). Este hay sido suplantado por el protocolo BGP (Border Gateway Protocol), actualmente en la versión 4. BGP puede enrutar paquetes de acuerdo a varias políticas. Estas políticas pueden estar basadas en entras políticas, seguridad o consideraciones económicas. Uno de los objetivos principales de BGP es recortar el monto de tráfico que no es local en este sistema autónomo.

Implementaciones UNiX

	RIP	Hello	OSPF	IS-IS	BGP
EGP					
routed		**X**			
gated v2	**X**		**X**		**vl**
gated v3.5	**X**	**X**	**X**	**X**	**vl,v2,v3,v4**

Históricamente en los sistemas UNIX, el demonio implementa enrutamienro Dinámico basado en RIP. Cuando comienza, routed inicializa la tabla de información de enrutamiento basado en el archivo /etc/gateways. El formato de este archivo es similar al argumento del comando route.

[net | host] dirección gateway gateway metric [active | passive]

El passive indica que el gateway no difundirá información y, por lo tanto, no serán sacado de la tabla de enrutamiento. Gateways activos, serán removidos de la tabla de enrutamiento si ellos no envían información de enrutamiento.

Los mensaje son transmitidos en cada red, advirtiendo sobre su tabla de enrutamiento cada pocos segundos. El programa también escucha los mensajes que envía desde otros programas routes y actualiza la tabla de enrutamiento local sin información recibida es considerar importante.

El daemon gated es un reemplazo moderno del daemon routed. El daemon de enrutamiento gated soporta mucho más protocolos. En particular, es en un protocolo de enrutamiento Dinámico y esta disponible brevemente.

Ejercicios 3-4: Protocolos de Enrutamiento Dinámico

En este ejercicio, trabajaremos con preguntas relacionadas con el tema de enrutamiento Dinámico para la Compañía X. No se proveen soluciones para este ejercicio.

1. RIP usa un broadcasts cada 30 segundos para anunciar las rutas disponibles y el número de saltos a ellas. En esta ruta, dos enrutadores están sirviendo tres redes y ambos enrutadores están ejecutando RIP (versión 1). Complete los detalles de las rutas anunciadas y los saltos después que el sistema ha normalizado sus operaciones.

2. Ahora si el enlace de R2 a la Red A se cae. ¿Qué le pasaría a las rutas y los saltos anunciados por ambos R1 y R2 en la reded B y C? ¿Qué más puede pasar?

3. El administrador de Red de Compañía X esta considerando si es apropiado usar enrutamiento Dinámico. El administrador está enfrentando la siguiente situación.
   La Compañía ha crecido de nuevo y ahora desea conectar otro sitio a su red corporativa. Para hacer esto, el administrador desea colocar otra conexión de 1-Mbps dentro de ambos sitios, así suplementando una conexión existente de 2-Mbps entre ellas. La red WAN debe verse entonces así: IMAGEN AQUI

La ventajas de este ordenamiento esta supuesto a ser resistente en el caso de fallas de uno de sus enlaces. Enrutamiento puede automáticamente cambiarse.

Usted debe tomar en consideración:

- ¿Vale la pena que la Compañía X considere ejecutar un protocolo de enrutamiento sobre los enlaces WAN?
- Y se es así, ¿cual protocolo(s) pueden ellos ejecutar?
- ¿Tiene la escogencia de un protocolo sobre otro algún beneficio?¿Cuáles son estos beneficios?
- ¿Existe algún beneficio en ejecutar enrutamiento Dinámico dentro de la Compañía X?

IP MULTICAST

IP Multicast es una manera efectiva para transportar paquetes desde un origen a multiple destinos. Esto se logra habilitando el origen a enviar paquetes a multiple recibidores Cuáles desean explícitamente recibirlo. Los receptores se hacen parte de un grupo multicast en particular y los paquetes son entregados a todos los miembros del grupo.

El proceso es eficiente por que las aplicaciones IP tradicionales emplean unicasting en la cual una conexión TCP es establecida entre un único emisor y cada uno de los receptores. El emisor replica la data sobre cada conexión, la cual puede usar muchos recursos de red. Aún más, el número de receptores esta limitado por el ancho de banda y el total desempeño se disminuye. El IP Multicast se asegura que las aplicaciones que requieren entrega a multiple destino no causen congestión en la red con copias múltiples de la misma data.

IP Multicast esta basado en el Protocolo UDP (User Datagram Protocol) y por esto, no es confiable ya que el provee un servicio no orientado a conexión o sin estado. Un datagrama multicast es entregado a todos los miembros de un grupo de hosts destinatarios con el mismo esfuerzo y confiabilidad que un datagrama regular del tipo unicast. No hay garantía que los datagramas enviados serán recibidos y no asegura que los datagramas sean recibidos en el orden en que fueron enviados.

Hay tres niveles para que un host participe en IP Multicast. Ellos son:

Nivel 0 Un host no puede ni enviar o recibir IP Multicast
Nivel 1 Un host puede enviar pero no recibir IP Multicast.
Nivel 2 Un host puede enviar y recibir IP Multicast.

Los kernels GNU/Linux de versiónes superior a la 1.1.80 soportan enrutamiento multicast a través de la opción de configuración multicast. Para que un host participe en IP Multicast con niveles 1 y 2, es necesario tener software que permiten al host enviar y recibir datagramas multicast. La membrecía de un host a un grupo es dinámica; los hosts pueden entrar o salir de un grupo en cualquier momento. No hay restricción en el lugar o número de miembros en un grupo de hosts. Un host puede ser miembro de más de un grupo a la vez. Todas estas opciones requieren el router multicast router use el protocolo IGMP (Internet Group Management Protocol) para comunicar la información de membrecía del grupo. El IGMP es un protocolo simple de bajo nivel que permite a los hosts registrarse y recordar el estado de cada grupo multicast que el host pertenece. Para tener multicast habilitado en un sistema GNU/Linux, IGMP debe ser uno de los protocolos IP.

Hay varios protocolos de enrutamiento multicast siendo desarrollado para la toma de decisiones multicast. Entre ellas se incluyen DVMRP (Distance Vector Multicast Routing Protocol), CBT (Core-Based Trees), M-OSPF (La extensión Multicast al Open Shortest Path First), y PIM (Protocol Independent Multicast). PIM esta dividido en dos protocolos: modo denso llamado PIM-DM y modo dispersado llamado PIM-SM. La función principal de este protocolo de enrutamiento es pasar la información de membrecía de grupo entre los

enrutadores multicast y enrutar los datagramas a cada miembro del grupo multicast.

Las función de enrutamiento multicast en un host GNU/Linux es proveído por el daemon llamado mrouted, cual recibe los paquetes unicast encapsulados en una interfase entrante, entonces enruta los paquetes sobre el conjunto de interfases salientes donde quiera que existe el host miembro. Este es similar al daemon routed, pero la diferencia es que routed usa el protocolo de enrutamiento RIP, en donde mrouted usa DVMRP para el enrutamiento multicast. Este daemon es requerido para el enrutamiento de tráfico multicast o proveyendo un tunel a otro sitio que explicamos más adelante.

La configuración de mrouted puede ser cambiado con la modificación del archivo de modificación /etc/mrouted.conf. Existen comandos diferentes para los requerimientos variados.

El comando phyint puede ser usado para deshabilitar enrutamiento multicast en la interfase física. Este comando debe presidir los comandos de tunel, Cuáles son usados para establecer el vínculo de tunel entre una dirección IP local y una dirección IP remota. La opción cache_lifetime determina el monto de tiempo que una ruta multicast capturada en cache permanece en el kernel antes de dar time out. El valor por defecto es de 300 segundos, pero puede ser cambiado a cualquier valor entre 300 (5 minutos) y 86,400 (1 día).

La opción pruning <off/on> es proveída para que mrouted actúe como un servidor de enrutador que no hace pruning. La configuración por defecto habilita a pruning. También es posible iniciar a mrouted en modo de no prunning usando la opción -p en la línea de comandos. Esto solo debe ser llevado acabo para el propósito de prueba solamente. La opción threshold es la mínima IP TTL requerida por un datagrama multicast para ser reenviada a una interfase dada o un tunel. Es usada para controlar el alcance del datagrama multicast y cada enrutador multicast reduce el campo TTL por 1.

La opción rate_limit permite al administrador de red especificar a cierto ancho de banda en Kbps que fueran asiganadas a tráfico multicast.

El comando mrouted responde a las siguientes señales:

HUP Esta reinicia mrouted y el archivo de configuración es leído nuevamente cada vez que la señal es invocada.

INT, TERM Ambos comandos terminan la ejecución correctamente notificando a todos los routers vecinos.

USR1 Este da volcado de las tablas internas a /usr/tmp/mrouted.dump

USR2 Este comando vuelca el cache interno de las tablas a /usr/tmp/mrouted.cache

QUIT Volca las tablas de enrutamiento a stderr si el mrouted fué invocado con un valor de debug mayor que cero.

El programa mrouted escribe su pid a /etc/mrouted.pid al inicio por conveniencia de Envío de señales. Este no afecta los protocolos de enrutamiento ya existentes. Un host que ejecuta el daemon mrouted lo hace además de los protocolos estándares de enrutmiento.

El protocolo de enrutamiento estándar ya existente efectúa un tunel multicast en casos donde los datagramas multicast deben pasar a través de routers intermedio que no soportan IP multicast. Para los hosts en redes separadas para comunicarse usando, los routers en ambas puntas deben configurar mrouted usando el comando tunnel. El tunel debe ser establecido en el archivo mrouted.conf de los dos routers que el tunel interconecta. Los routers en ambas puntas del tunel entonces escucha por datagramas que fueron enviados al grupo multicast. Cuando el datagrama multicast arriba en un router pero su destino es el otro router en la otra punta del tunel, mrouted envía el datagrama multicast a mrouted al otro router usando una dirección IP unicast, la cual encapsula el datagrama multicast. En la otra punta del tunel, mrouted extrae el datagrama multicast desde la

dirección IP unicast encapsulada y usa la dirección de destino multicast para reenviar el datagrama a su destinación final.

RESUMEN

En este capítulo, fueron introducidos los siguientes tópicos:

- Enrutar (Routing) y los tipos de protocolos de enrutamiento
- Subnetting y supernetting
- Clases de Redes de IP y como sobre pasar sus limitaciones
- Como dividir una red en multiple subnets
- Como incrementar el número de redes y reducir el número de hosts
- Transporte/Delivery de Paquetes
- El rol de las Tablas de Enrutamiento
- Enrutamiento Dinámico
- Filtrado de Paquetes y Firewalls (Corta Fuegos)

- Multicasting

PREGUNTAS POST-EXAMEN

Respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿En que se diferencian los conceptos de supernetting del de subnetting?
2. ¿Por qué es que no podemos usar direcciones que terminen en 0 o 255 como direcciones de una subnet?
3. Considerando RIP y OSPF ¿Cúal de estos protocolos de enrutamiento es más avanzado y por qué?
4. ¿Después de cuantos saltos considerará RIP un destino no alcanzable?



TCP, UDP, ICMP y la Capa de Transporte

Objetivos

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Instalar redes TCP/IP y definir los puertos comunes.
- Identificar los protocolos usados para establecer conexiones entre nodos interconectados asi como las convenciones comunes usadas por cada protocolo.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué significa IP?

2. ¿Es el encabezado de IP una fuente principal del sobre peso en la mayoría de implementaciones de IP?

3- ¿Cuáles tres campos son contenidos en el cabezal IP?

4. ¿Cuáles números de puertos son reservados para servicios bien conocidos?

INTRODUCCIÓN

En este capítulo cubriremos la capa de Transporte de la pila del protocolo TCP/IP. Esto incluye el TCP (Transmission Control Protocol), el cual maneja las comunicaciones orientadas a conexión y el UDP (User Datagram Protocol), el cual maneja la comunicación sin conexión. El ICMP (Internet Control Message Protocol) es usado como una capa de control para permitir tráfico IP entre los sistemas para fluir libremente. También cubriremos formatos de tramas y fragmentación.

PROTOCOLO DE INTERNET

El Protocolo de Internet (IP) es un protocolo sin conexión, orientado a datagramas y reenviados de paquetes. La Data se pasa a IP desde los protocolos más alto de transporte para entrega. El IP envía un paquete en la próxima etapa de su viaje hacia su destino final basada en la dirección IP de destino y la información de enrutamiento que se encuentra almacenada internamente dentro del protocolo. Un paquete IP puede que pase por varios gateways o saltos, antes de alcanzar su destino final.

A menudo el IP es describido como un protocolo no confiable. Esto no significa que no sea útil. Sino, esto significa que su entrega no es garantizada, IP depende de entrega del mejor esfuerzo; los paquetes en esta capa no son confirmados.

El hecho que IP no garantiza la entrega no es de ninguna gran preocupación. Ante todo, IP asume que las capas más altas (como es la de la capa de Transporte) serán las responsables de proveer la confiabilidad. Segundo, IP fué diseñado para ser simple, eficiente y fácil de implementar. Se da muchas veces el caso que todo lo que es requerido es que sea rápido fast y la entrega de mejor esfuerzo.

Piense en no confiable al igual que enviar un correo tradicional. Una vez enviado, usted no puede estar seguro si será entregado a su destino final o si llegará en su totalidad. Pero sabemos, que basada en la confiabilidad de del servicio postal haya un buen chance de que nuestro correo llegue a su destino y que el tiempo que tome dependerá de la distancia que nuestro correo tenga que viajar (y la eficiencia de cada una de las diferentes agencias postales que nuestra carta tendrá que viajar para llegar a su destino). Asi que, el servicio postal e un mecanismo de entrega basado en mejor esfuerzo, y mecanismos de entrega no confiable, pero este no es un problema mayor que no nos permita usarlo y esta basado en el nivel de confiabilidad que este provee.

Si estas inseguro de como se puede proveer un servicio confiable en un nivel más alto que usa IP (un servicio no confiable) en un nivel más bajo, consideremos como podemos operar un servicio de transferencia de data que garantice confiabilidad usando el servicio postal. Una idea que surge es la de establecer un mecanismo en el cual se establezca un método de confirmación de devolver una postal al emisor original. Si usted envía una tarjeta y después recibe una confirmación para esa tarjeta, usted puede estar seguro que la original llegó con éxito.

Los Protocolos

El sistema TCP/IP esta basado en un conjunto de protocolos en capas. Esto permite implementaciones en cada capa ser separadas de cada una esto hace que el sistema completo sea más modular y más fácil de depurar.

El TCP/IP contiene dos protocolos básicos para la transmisión de data estos 2 protocolos son UDP y TCP, en el modelo OSI, ellas residen en la capa de transporte. Además, existen el ICMP, el cual se encuentra en la capa de red.

ICMP

Localizado en la capa de red del modelo OSI es el protocolo de control de mensajes del Internet (Internet Control Message Protocol). El propósito del ICMP es proveer un mecanismo de control de error al IP. Los mensajes ICMP son de dos tipos básicos: Control de error y consulta. Algunos mensajes ICMP de control de error son:

3	Destino inalcanzables
4	Source Quench
5	Redirección
11	Tiempo Excedido
12	Problema de Parámetro

Los mensajes ICMP usado para consulta son:

0	Echo Reply/Repuesta
8	Echo Request/Petición
9	Router Advertisement/Anuncio
10	Router Solicitation/Solicitud
13	Timestamp Request/Petición
14	Timestamp Reply/Repuesta
17	Address Mask Request/Petición de Mascara de Dirección
18	Address Mask Reply/Repuesta de Mascara de Dirección

UDP

El Protocolo de Datagrama de Usuario (User Datagram Protocol) es un protocolo de la capa de Transporte utilizado principalmente para comunicaciones cortas y de una sola vía. Los datagramas UDP no contienen información acerca de los otros paquetes. Asi que, cada paquete UDP es independiente uno del otro, cada mensaje no puede ser mayor en tamaño al máximo permitido por el paquete UDP.

Otro aspecto del UDP es su falta de control de error. La única forma de implementar algún tipo de control de errores es a traves del uso de checksum. Si el checksum no computa apropiadamente, el paquete es dejado caer, el UDP depende del ICMP para enviar los mensajes apropiados de error.

Puede ser que el UDP no le parezca muy útil, pero, este provee una magnifica alternativa para las comunicaciones que sólo necesitan enviar pequeños mensajes sin el alto costo fijo de las comunicaciones a gran escala usada por el TCP.

TCP

Debido a la falta de funcionalidad ofrecida por el protocolo UDP, para las comunicaciones que requieren más confiabilidad, el TCP (Transmission Control Protocol) es usado. TCP es un protocolo más robusto que el protocolo UDP; este provee control de error, flujo de data y comunicación de dos vía.

Las conexiones TCP administran las comunicaciones de dos vía y manejan errores a traves de opciones (flags) en el cabezal del paquete TCP. Las dos opciones más comunes del TCP son sincronizadas (SYN) confirmación (ACK). SYN es enviado al principio de una conexión y ACK es enviado una vez el paquete ha sido recibido por el host destinatario.

Una conexión empieza con el equipo cliente enviando una información de inicializacion la máquina servidor en un paquete con la opción SYN establecida. El servidor entonces reconoce la información de inicialización del cliente con una opción ACK y envía su propia información de inicialización con la opción de SYN establecida. El cliente entonces reconoce los paquetes SYN ACK del servidor con otro paquete ACK. Este proceso de inicialización es lo que es conocido como el saludo de tres vía (three-way handshake).

El control de error es implementado en el TCP a traves de la opción ACK. cada paquete enviado por TCP es enumerado y un temporizador es iniciado cuando cada paquete es enviado. Si una confirmación no es recibida antes de que se venza el temporizador, el paquete entonces es reenviado.

El flujo de data es acomodado a traves del cabezal del paquete. Como TCP enumera cada paquete, el número del paquete es colocado en el cabezal del paquete TCP. El emisor entonces, puede enviar multiple paquetes consecutivamente. Cuando cada paquete arriba, este es colocado en la cola y entonces los paquetes recibidos pueden ser colocados en el orden apropiado.

FORMATO DEL PAQUETE IP

El campo Header Length (Longitud del Cabezal) específica la longitud del cabezal (header). La longitud del header se mide en words de 32-bit, dando un tamaño máximo del cabezal de 60 bytes. Por lo general el Header Length es 5.

El campo Versión lleva la versión actual del IP, la cual hoy es la 4. La próxima versión de IP, la versión 6, será la próxima generación IPng, esta actualmente siendo actualmente especificada.

El campo Service Type (Tipo de Servicio) es usado por algunos enrutadores al tener que decidir como reenviar el datagrama. Este puede indicar que el datagrama debe ser enviado via el router más barato, rápido y más confiable o por de más ancho de banda. Un datagrama puede ser enviado sin la opción de Service Type establecida.

La Total Length (Longitud Total) es el tamaño total del datagramas en bytes, incluyendo el cabezal. Esto nos da el tamaño máximo del datagrama de 65,536 bytes, o 64 KB.

Los campos Identification, Flags y Fragment Offset son usados para controlar, si es necesario, la fragmentación y el reemsamblaje de los datagramas. De esto hablamos más adelante en este mismo capítulo.

El campo Time to Live (Tiempo Restante) es un contador que es reducido cada vez que un paquete cruza el gateway. Cuando el contador llega a ‘0’, el paquete es descartado. Esto es para prevenir que datagramas que están equivocadamente enrutados congestionen la red si viven un tiempo indefinido.

El campo Protocolo (Prorocol) indica cual protocolo de transporte es que va a recibir el paquete en la dirección de destino. Todos los protocolos que usan IP (TCP, UDP, ICMP, etc.) llevan un checksums de su propio cabezal y data. Asi pueden evitar redundancia, el Header Checksum cubre solamente el cabezal IP. Si la capa de IP receptora detecta un datagrama corrompido, este lo descartará.

El datagrama será entregado a la dirección especificada en el campo ‘Destination Address’ del encabezado. Si este no se encuentra en la misma red como el sistema actual, entonces el datagrama es reenviado a un gateway apropiado que le dará el destino final. Esta información de enrutamiento es almacenada en una tabla por separado. Además, la dirección de origen ‘Source Address’ del datagrama esta incluida en su propio campo.

Las Opciones IP pueden ser usadas para gobernar el enrutamiento (enrutamiento estricto o liviano) de los datagramas, establecer cualquier restricciones de seguridad y manejo o grabar las direcciones y la fecha de los enrutadores por los cuáles los datagramas pasaron para arribar a su destino final. Algunos enrutadores no examinan estas opciones.

En esta sección discutiremos los siguientes tópicos:

- IP Checksum

- Tipo de Servicio
- Fragmentación y Reemsamblaje
- Tiempo de Vida/Time to Live
- Opciones IP

Calcular el cheksum del encabezado IP es causa de costos considerables en las implementaciones IP. Por estas causa es que la nueva versión de IP, la versión 6, ha removido el checksum. ¿Cómo, entonces, podemos garantizar la integridad del encabezado IP? (Cubriremos este tópico más adelante.)

Tipo de Servicio

Los bits del campo de tipo de servicio (TOS) son un conjunto de cuatro indicadores de un bit de la cabecera de IP. Si uno de estos indicadores de bit vale 1, un encaminador puede manipular el datagrama de forma diferente del caso en el que ningún indicador valiera 1. Cada uno de los cuatro bits tiene un propósito diferente y sólo uno de los bits de TOS puede valer 1 al mismo tiempo, es decir, las combinaciones no están permitidas. Estos indicadores de bit se denominan de "tipo de servicio" porque permiten que la aplicación que transmite los datos informe a la red del tipo de servicio de red que requiere. Los bits del ToS le permiten a la máquina que envía requerir que los enrutadores manejen los paquetes de cierta manera. Una manera posible es precedencia. Esto permite para la prioridad de paquetes, aunque por lo general no es implementado en los enrutadores. Típicamente, ToS permite que un paquete sea enrutado basado en uno de lo siguiente:

Demora mínima/Minimal Delay

Se utiliza cuando se le da la máxima importancia al tiempo de viaje de un datagrama del 'host' de origen al 'host' de destino (demora). Por ejemplo, un suministrador de red podría estar utilizando tanto conexiones de red de fibra como por satélite. Los datos transportados por las conexiones por satélite tienen que viajar más lejos y su demora entre los mismos extremos será por lo general mayor que la de las conexiones de red terrestres. Un suministrador de red podría elegir asegurarse que los datagramas con este tipo de servicio no se transporten por satélite.

Rendimiento máximo/Maximal Throughput

Se utiliza cuando el volumen de datos transmitidos en cualquier período de tiempo es importante. Existen numerosos tipos de aplicaciones de red para las que el tiempo de demora no es muy importante pero el rendimiento sí que lo es; por ejemplo, las transferencias de ficheros en bloque. Un suministrador de red podría elegir encaminar los datagramas con este tipo de servicio vía rutas de demora alta, pero de gran ancho de banda, como las conexiones por satélite.

Fiabilidad máxima/Maximal Reliability

Se utiliza cuando es importante tener alguna certeza de que los datos llegarán al destino sin necesidad de una retransmisión. El protocolo IP puede transportarse sobre un número variado de medios de transmisión de bajo nivel. Mientras que SLIP y PPP son adecuados para protocolos de enlace de datos, no son tan fiables para transportar IP como otras redes, como las redes X.25. Un suministrador de red podría tener disponible una red alternativa, que ofreciera alta fiabilidad para transportar IP y que se utilizaría cuando se eligiera este tipo de servicio.

Coste mínimo/Minimal Cost

Se utiliza cuando resulta importante minimizar el coste de los datos transmitidos. El alquiler de ancho de banda de un satélite para una transmisión transoceánica cuesta generalmente menos que el alquiler de espacio de un cable de fibra óptica sobre la misma distancia, por lo que los suministradores de red pueden elegir proporcionar ambos y cobrar de forma diferente según sea el que se utilice. En este escenario, el bit de tipo de servicio de "coste mínimo" puede ocasionar que los datagramas sean encaminados vía la ruta de menor coste por satélite.

Fragmentación y Reensamblaje

Cuando la capa de Transporte pasa los data al IP, esta será organizada dentro de un datagrama IP. Si el datagrama creado es más grande que el MTU, Unidad Máxima de Transmisión (Maximum Transmission Unit), de la red, el IP dividirá la data en partes más pequeñas llamadas fragmentos. Estos fragmentos son encapsulados en en paquetes más pequeños, los cuáles son enviados por la red al host receptor y

reensamblado.

Como cada fragmento es un paquete IP individual, cada uno de estos paquetes puede ser enrutado a su destino cía tres rutas diferente. Esto significa que al único host que todos los fragmentos son garantizados a arribar es el host receptor. Asi que, el reensamblaje sólo toma lugar en el host de destino.

La fragmentación original toma lugar basada en el MTU del host que envía. A medida que los fragmentos son enrutados a su destinación, el MTU puede puede variar de una red a la otra. Esto quiere decir que si un datagrama que contiene un fragmento confronta una red con un MTU inferior, el fragmento mismo puede ser que sea aún más fragmentado. Esto se maneja bien por IP debido a que cada fragmento contiene información de como (offset) recomponerse del principio del datagrama original.

Si la capa de Transporte desea asegurarse de que un datagrama no este fragmentado, le puede instruir a IP establecer la opción (flag) de NO fragmentar en el cabezal del datagrama. Si el cualquier punto del viaje cualquier datagrama es más grande que MTU, este será descartado. En este caso, ICMP causará que un mensaje de error sea generado.

Aquí le presentamos un ejemplo de fragmentación:

El cabezal IP contiene tres campos para asistir con el proceso de fragmentación: Identification/

Host A	4,000 byte UDP datagrama Ethernet: MTU - 1,500 bytes de data			**Host B**
	Datagrama Original de IP	**Fragmento 1**	**Fragmento 2**	**Fragmento 3**
Identification	47987	47987	47987	47987
Fragment Offset	0	0	1480	2960
Size	4,000	1480	1480	1040
Flags	None	More Frags	More Frags	none

Identificación, Fragment Offset/Posición del Empalme y Flags/Opciones.

Cada datagrama IP tiene un campo único de Identificación dentro de su cabezal. Este es un campo de un número de sólo 16-bit mantenido por el host emisor, el cual es incrementado por cada datagrama que este envía (sin incluir los fragmentos). Si un datagrama está fragmentado, cada fragmento compartirá el número de Identificación original del datagrama. Esta es la forma por la cual el host receptor sabe a cual datagrama en particular pertenece un fragmento.

El Fragment Offset ofrece el punto de empalme desde el datagrama de origen al fragmento actual. Asi es que el receptor sabe donde cada fragmento cae en el datagrama original al momento de reemsamblaje.

El campo Size/Tamaño de cada fragmento contiene el tamaño del paquete. La suma del tamaño del total de todos los fragmentos será el tamaño del datagrama original. El campo Size muchas veces también es referido como el campo Longitud Total (Total Length).

haya una opción/flag en particular establecida en el campo Flags para todos los fragmentos excepto el último. La opción more fragments es de la manera que el receptor reconoce de esperar por más fragmentos

antes de empezar a reensamblar el datagrama. A primera vista aparenta que el último paquete de fragmento pudiese ser confundido con un datagrama normal, un datagrama sin fragmentar. Pero, el software IP receptora encontrará que el Fragment Offset no es igual a cero y por eso identificará el último fragmento del datagrama. Esto es importante porque los fragmentos pueden ser enrutados por separado, los fragmentos no siempre arribaran en el mismo orden en el cual fueron enviados.

Si todos los fragmentos de un datagrama no son recibidos dentro de un período de tiempo, normalmente 30 o 60 segundos, el fragmento recibido será descartado, igual que si el datagrama nunca hubiese llegado. No haya manera de reenviar solamente un fragmento de un datagrama. Si el fragmento fué producido por un enrutador intermediario, el host que envía ni sabe si el datagrama que el el envía es fragmentado o no. Esto significa que el datagrama por completo deberá ser retransmitido por completo. Por esta razón, la fragmentación es muy amenudo evitado.

Los protocolos de la capa de Transporte manejan la fragmentación en diferente maneras. El TCP trata de evitar la fragmentación reduciendo el tamaño de sus segmento. En algunas aplicaciones UDP, es posible limitar el tamaño de los datagramas pasados a IP.

Tiempo de Vida/Time to Live

Específica el tiempo(en segundos) que se le permite viajar a este datagrama. Cada "router" por el que pase este datagrama ha de sustraer de este campo el tiempo tardado en procesarlo. En la realidad un "router" es capaz de procesar un datagrama en menos de 1 segundo; por ello restará uno de este campo y el TTL se convierte más en una cuenta de saltos que en una métrica del tiempo. Cuando el valor alcanza cero, se asume que este datagrama ha estado viajando en un bucle infinito y se desecha. El valor inicial lo debería fijar el protocolo de alto nivel que crea el datagrama.

Bajo ciertas circunstancias, como es por ejemplo un enrutador mal configurado, los paquetes pueden potencialmente viajar en un bucle para siempre. El efecto de este mecanismo puede ser visto desde el utilitario traceroute.

Opciones IP

Las Opciones de IP permiten al host emisor requerir cierta condiciones de enrutamiento o otros servicios de los routers. Estudiaremos a cuatros principales: Record Route, Timestamp, loose-source routing, y strict-source routing. Al usar la opción Record Route esta nos permite insertar la ruta que el paquete viajó dentro del cabezal IP, escribiendo en secuencia la direcciones IP de todos los routers que el paquete atravesó.

Desafortunadamente esto no es tan útil como se ve, ya que haya in espacio limitado en el cabezal para almacenar estas rutas. Puesto en práctica, para almacenar la ruta de los paquetes, debe usar un mecanismo basado en TTL similar al de traceroute. haya equipos de enrutar que no soportan Record Route.

En su viaja a traves del Internet, un paquete puede cruzar 20 y hasta 30 routers desde su dirección de origen hasta su destino final. Esto excede por mucho la capacidad del mecanismo de Record Route.

Timestamp/Sello de Tiempo

El Timestamp fué concebido como una extensión del mecanismo de Record Route. La hora y la fecha que un paquete pasa a traves de un router es almacenado, además de su dirección IP. Desafortunadamente, esto reduce el espacio disponible aún más, asi que el Timestamp es aún más limitado que Record Route.

peor aún, el tiempo que es sellado es el del router local. El tiempo entre dos routers no necesariamente

tiene que estar sincronizado, al menos que un protocolo como el Network Time Protocol (NTP) se este ejecutando. Asi que los tiempos reportados de retraso pueden ser no actualmente ciertos sino afectado por falta de sincronización. Vuelvo y repito, esta opción no siempre esta disponible en los enrutadores más modernos por falta de popularidad de su uso.

Source Routing/Origen del Enrutamiento

También es posible especificar por adelantado que ruta debe seguir un paquete. Esto se puede lograr de dos maneras:

- **Loose-source routing/Enrutamiento Estricto**

  Aquí le especificamos en el cabezal del IP una lista de enrutadores por los cuáles el paquete deberá atravesar. El paquete también puede viajar vía otros routers que no estén listados en la lista pero sólo será exitoso en alcanzar su destino final si lo hace vía esos routers listados.
- **Strict-source routing/Enrutamiento Flexible**

  En este escenario la lista en el cabezal explícitamente específica cuáles enrutadores el paquete deberá atravesar. No es permitido viajar vía cualquier otro enrutador que no este listado.

Enrutamiento Flexible y Enrutamiento Estricto (Loose- and strict-source routing) fueron conce para proporcionar una medida de control sobre la ruta que los paquetes toman. Pero, como las otras opciones, estas on muy dificiles de poner en practica. Muchos enrutadores modernos se reusan aceptar paquetes con las opciones de source-routed aplicadas.

En la practica, el control del enrutamiento está basado en el uso de políticas de enrutamiento dinámico (presentaremos más información sobre esto más adelante) y la cooperación mutua entre los administradores de sistemas de las diferentes organizaciones que el tráfico de sus paquetes transversan.

MANEJO DE ERROR ICMP

El IP no tiene mecanismo interno del manejo o reporte de errors. Aunque esto tiene como ventaja que el protocolo es más simple, esto significa que los errores enfrentados con el transporte y enrutamiento no serán anunciados y permanecerán sin ser corregidos. El ICMP y el IP son virtualmente lo mosmo excepto por unas cuantas pequeñas diferencias:

- ICMP agrega el reporte de error y otros mensajes de control a IP.
- ICMP también reside en la capa de Red y trabaja en conjunto con el IP. Usa el IP como agente de transporte.
- Los mensajes ICMP son usados para indicar errores como cuando los hosts no son alcanzables, tiempo de espera agotado o timeouts, o error en el formato del datagrama. También pueden ser usados para el control del flujo cuando una destinación nos informa que no puede aceptar más datagramas (source quench). El ICMP puede ser usado para requerir información como son los timestamps (para la sincronización del reloj) o simplemente una respuesta a un petición de echo (echo es usado para probar conexiones, como en el comando ping).
- Con la excepción de las peticiones y respuestas de echo del ICMP, los mensajes ICMP no son enviados en respuestas a otros mensajes ICMP. Ellos tampoco son enviados en respuestas a los datagramas de respuestas. Esta es una característica importante que evita la posibilidad de tormentas de broadcast, en la cual mensajes son intercambiados entre los hosts, lo cual consumirá mucho ancho de banda.

Los siguientes tópicos son discutidos en esta sección:

- Formato del Cabezal ICMP
- Tipos de Mensajes ICMP

Formato del Cabezal ICMP

Cada mensaje ICMP es encapsulado en un datagrama IP. El mensaje tiene cuatro partes:

- Los tipos de mensajes. Existen quince tipos diferente de mensajes ICMP, descritos más´s adelante en esta

misma sección.

- El código del mensaje es en efectivo un subtipo. Por ejemplo, haya un tipo para el error “destination unreachable”. Existen dieciséis valores de códigos diferente para este tipo, las cuáles son establecidas por el emisor para decirle al receptor porque el paquete no pudo ser entregado.
- El checksum cubre ambos el cabezal y la data.
- La data difiere dependiendo en el tipo y el código del mensaje en el datagrama. Por ejemplo, todos los mensajes de error contienen el cabezal IP del datagrama erráneo y las primeras 8 bytes de su data, la cual contendrá parte del cabezal del protocolo del datagrama enviado. Este contiene el número del protocolo para que el receptor del mensaje ICMPsepa a quien informar del error.

Tipos de Mensajes ICMP

Los quince tipos de mensajes ICMP son como sigue:

Tipo	Uso
8	Petición de Echo (enviado por ping)
0	Repuesta de Echo (recibido por ping)
3	Destino inalcanzable (no puede entregar o reenviar un datagrama)
4	Control de flujo (Source quench)
5	Redirección (alterar tablas de enrutamiento en el receptor)
9	Anuncio de Router
10	Solicitud del Router
11	Tiempo Excedido (TLL igual a 0)
12	Parameter problem (invalid option in IP header)
13	Petición del Sello de Tiempo/Timestamp (usado para sincronizar de la red)
14	Respuesta al Sello de Tiempo/Timestamp (usado para sincronizar de la red)
15	Petición de Información (obsoleto)
16	Respuesta de Información (obsoleto)
17	Petición de mascara de dirección
18	respuesta de mascara de dirección

TCP Y LA CAPA DE TRANSPORTE

En esta sección cubriremos las capas encima del IP (Internet Protocol). Discutiremos el primero de dos protocolos de transporte del TCP/IP- Transmission Control Protocol (TCP). En esta sección también se incluyen algunas aplicaciones que usan este protocolo, el concepto de sockets y número de puerto y el mapeo de estos números de puertos a servicios.

Los siguientes tópicos son discutidos en esta sección:

- ¿Por qué tener aún más Capas?
- Interfase a la Capa de Transporte
- Conceptos
- TCP Fundamentos
- TCP Formato de la Trama
- TCP Opciones
- TCP Control de Flujo
- SYN Cookies
- Puertos y Sockets
- Pasar Data a Aplicaciones
- HTTP
- Servidores GNU/Linux

¿Por qué tener aún más Capas?

El IP es excelente en lo que hace, pero este fué diseñado para un propósito específico y no ofrece todo lo que necesitamos para un ambiente de aplicaciones distribuidas en red. Aquí le presentamos una lista de las piezas que aún faltan para completar el rompe cabeza:
- Una manera de conversar con una aplicación específica en un host
- Un mecanismo confiable de transferencia de data
- Una manera de enviar flujo continuo de data
- Control de Flujo de data

Pero, afortunadamente estas piezas que faltan son exactamente lo que la capa de Transporte fué diseñada a enfrentar. Ahora podemos observar alguno de los temas y maneras que esto puede ser resuelto.

Interfase a la Capa de Transporte

El IP es usado por los dos mayores protocolos de Internet de la capa de Transporte, el TCP y UDP, para transmitir data sobre la red. La porcion de data de un datagrama no contiene indicación de cual protocolo se encuentra operando. Pero, el cabezal IP si contiene un campo Protocol Number (Número de Protocolo) que es usado para especificar cual protocolo de transporte he usado para enviar la data. En el destino final, el IP puede pasar data al software apropiado del protocolo de transporte.

El número del protocolo es también usado para identificar los paquetes ICMP y lo reenvía al software del protocolo ICMP para procesarlo, aunque este no es estrictamente un protocolo de transporte.

Conceptos

Todo lo que la capa de Red provee es la transmisión de paquetes de IP de host a host a traves de la red. Esta no tiene conocimiento alguno de cuáles aplicaci son propietarias de cual data pero simplemente se le canalizan toda la data para un host en particular de punta a punta en la red.

En la capa de Transporte es donde se comienza a fragmentar la data y a direccionarla a y desde las aplicaciones a traves de la red. Como veremos en esta sección la próxima, existen dos tipos de flujos disponibles: los confia y los no confiables.

Protocolos sin Conexión

Imaginemosnos las tramas de data como una carta postal. Durante su viaje por diez ciudades, usted envía una carta postal de cada ciudad que visita a un amigo en otra ciudad. El envío es hecho a traves de un protocolo no confia (como es el servicio de correo) lo que significa que no haya garantía alguna que las diez cartas llegarán a las manos de su amigo en la otra ciudad. Aunque el servicio no es garantizado por lo general las cartas enviadas por el servicio postal llegarán a su destino final, tampoco no haya garantía de que llegaran en el orden que se enviaron.

El IP trabaja de esta misma manera. Los paquetes IP pueden perderse o hasta dañarse. No haya un mecanismo en la capa de Red para manejar las notificaciones de la perdida al emisor o para requerir que el paquete sea reenviado. Estas funciones se dejan a un protocolo de más alto nivel.

Existen maneras de combatir este problema si se requiere oriniento correcto y confiabilidad. Por ejemplo, podemos asegurarnos que nuestro amigo en la otra ciudad envíe una carta de reconocimiento de que recibió nuestra carta. Entonces, un tiempo acordado podemos reenviar cualquier carta que no recibimos una carta de confirmación de el. Una vez se han recibido todas las cartas de confirmación, ya sabemos que todas nuestras cartas fueron recibidas.

También, podemos enumerar las cartas cuando las envió y entonces nuestro amigo puede usar estos números para ponerlas en orden y reconstruir todo el viaje por las ciudades de las diez postales.

Eso si, ambos de estos métodos tienen presentan problemas. Ahora cada vez que envío data debo esperar por una confirmación . La pregunta entonces es, ¿Qué tiempo debo esperar? Todo este mecanismo restará velocidad al proceso de la transmisión de la data. Si la transferencia es poca, funciona, pero para transferencias de data a alta escala, necesitamos un mecanismo diferente.

Protocolos Orientados a Conexión

Los protocolos Connection-oriented establecen una conexión con el otro punto, transfieren la data, luego cierran la conexión. La conexión en dos vía (full duplex) es analoga a la conversación por el teléfono. Los protocolos orientados a conexión son conocidos como confiable porque ellos incluyen una garantía de entrega de la data, asi previniendo la perdida de información y la data llega al otro punto en el mismo orden en que fué enviada.

Considere el intercambio de data como una conversación telefónica. Primero, la conexión es establecida, e.j., se marca el número y se conecta. Entonces el intercambio de la data en ambas direcciones comienza. Todo el tiempo, la data arriba al otro punto en el mismo orden en que fué enviada; esto es que sus palabras no llegan en un orden diferente al que usted la habló. Cuando la transferencia (en el ejemplo la conversación) es completada, la conexión es cerrada; y ambos cuelgan.

Este mecanismo es completamente confiable ya que no haya manera de perdida de data sin nuestro conocimiento. Este mecanismo es el que necesitamos en la transferencias de data a gran escala.

Fundamentos de TCP

El primer protocolo de transporte en el suite TCP/IP es el TCP. El TCP es muy sofisticado, garantizando que la data es entregada de punta a punta en una manera completamente confiable, en orden y sin duplicación. El TCP es un protocolo orientado a conexión. Lo que significa que una conexión logical, a veces conocida como un circuito virtual, debe ser establecido entre los dos puntos antes de que la data pueda ser transmitida o recibida. Esto es muy parecido a una llamada telefónica en la que uno debe marcar el número de teléfono de otra persona y alguien debe responder el otro teléfono antes de que podamos empezar a conversar.

Una vez establecida la conexión, el TCP garantiza que la data será entregada a su destino sin perdida ni duplicación. Cada porción de data, llamada un segmento, es transmitida con un checksum que puede ser utilizada para asegurarse que la data arrió intacta. El emisor espera que cada porción de data será confirmada su entrega y retransmitirá si no recibe una confirmación en un período acordado. Cualquier paquete que ha tomado más de 1,500 más para arribar a su destino final es considerado sospechoso y es descartado. Además, un paquete que ha efectuado más de 16 saltos o hops es considerado no confiable. Desafortunadamente, el Internet ha crecido tanto que es muy difícil encontrar un paquete que ha sobrevivido más de 10 saltos y que un paquete tome los 16 saltos permitidos es muy común.

El tamaño de un segmento es determinado por el software del TCP. Normalmente, una opción optima es hecha de acuerdo con la red subyacente. Por ejemplo, la elección para Ethernet es de 1,024 bytes ya que esta permite que cada segmento quepa tramas Ethernet individuales.

El TCP también permite el control de flujo fuera del mecanismo primitivo de source quench del ICMP a traves del uso del protocolo de sliding window. (lo veremos más adelante en este mismo capítulo en Control del Flujo TCP).

El TCP proveé una vista basada en flujo de bytes de las aplicaciones de comunicación. En el punto de recepción, el software TCP reensambla los segmentos en el flujo de data y lo pasa al software al nivel de aplicación. No existe fronteras lógicas en la data que se envía o l aquea se recibe, a diferencias del UDP, el cual

recibe y envía unidades discretas.

Formato de la Trama TCP

El TCP soporta full-duplex, comunicación orientada a flujo de bytes entre los procesos. Pero para la transmisión, los flujos de data deben ser divididos en segmentos que puedan ser encapsulados en datagramas IP. A cada segmento se le adhiere información de control y es entonces transmitida en un datagrama IP.

El cabezal TCP contiene varios campos importantes, incluyendo:

- **Source Port (16 bits)**
  Este campo identifica el puerto de origen. Cuando un cliente hace un petición a un servidor, el puerto de origen puede ser visto como el número de puerto de la aplicación cliente para la conexión inicial.
- **Destination Port (16 bits)**
  Este campo identifica el puerto de destino. Cuando un cliente hace un petición al servidor, el puerto de destino puede ser visto como el número de puerto de la aplicación servidor para la conexión inicial.
- **Sequence Number (32 bits)**
  Este campo indica el número de secuencia del primer byte de data en el segmento excepto cuando esta presente la opción SYN (este flag que hace referencia a un número sincronizado de secuencia). Si SYN esta presente, el número de secuencia es seleccionado al azar. El emisor y el receptor sincronizan el número de secuencia durante el establecimiento de la conexión.
- **Acknowledgment Number (32 bits)**
  Esta es una confirmación adicionada que contiene el número de la secuencia del próximo octeto que la entidad TCP está a la espera de recibir.
- **Header Length (4 bits)**
  Este campo específica el número de palabras de 32-bit en la cabecera. El tamaño máximo del cabezal TCP es de 60 bytes, pero el tamaño normal, sin las opciones es de 20 bytes. El cabezal TCP no contiene un campo Length. La longitud del segmento es calculado desde la longitud el datagrama IP y el HLEN TCP.
- **Reserved (6 bits)**
  This field is reserved for future use.
- **Flags (6 bits):**
  - **URG** el puntero urgente. Indica que data ha sido colocada en e flujo de data.
  - **ACK** es una confirmación. Identifica información de la confirmación en el packet.
  - **PSH** es la función push. Forza a TCP a liberar la data.
  - **RST** reinicia la conexión. Rápidamente desactiva una conexión TCP.
  - **SYN** sincroniza un número de secuencia. Este bit se establece en 1 para los primeros dos paquetes para las conexiones que usan TCP en la capa de Transporte.
  - **FIN** significa que el emisor no transmitirá más data. Es la manera normal de finalizar una conexión TCP.
- **Window (16 bits)**
  Este es el asignador de crédito al control de flujo, en bytes. Contiene el número de octeto de data, empezando por el indicado en el campo Acknowledgment, que el emisor está dispuesto a aceptar.
- **Checksum (16 bits)**
  Este es usado para la detección de error. El checksum convierte la data en segmento, su cabecera y también las direcciones de origen y de destino y el valor del protocolo desde la dirección IP. En una manera similar al UDP, si el receptor detecta que el datagrama se ha corrompido, este lo descarta. Pero, el TCP posee un mecanismo interno para el reenvio de data que no ha sido confirmado. Esto asegura que la data será entregada definitivamente.
- **Urgent Pointer (16 bits)**
  Este campo apunta al byte después de la urgent (urgente) data. Asi que, el receptor puede determinar cuanta data urgente está incluida en el mensaje.
- **Option Type (8 bits)**
  Actualmente una sola opción, la opción 2, está definida. Específica el Tamaño máximo de Segmento (Maximum Segment Size, MSS) para la conversación.
- **Option Length (8 bits)**
  Este campo específica la longitud de la opción. La opción 2 (MSS) es de 4 bytes de largo.
- **Maximum Segment Size (MSS) (16 bits)**
  El tipo de opción más común del TCP, específica el tamaño de segmento más largo que el TCP transmitirá al otro node. La información MSS es intercambiada durante el establecimiento de la sesión TCP.

TCP Handshaking/Saludo

Las seis opciones/flags de TCP involucran varios mensajes de control que son enviados para poder establecer la conexión y cerrarla. Estas opciones tienen los siguientes significados:

SYN Número de secuencia sincronizado para iniciar la conexión
ACK Confirmación/Acknowledgment de la trama
RST Reinicia la conexión
FIN El emisor ha acabado de enviar la data
URG Data entrante es Urgente
PSH Pasar esta data a la aplicación lo más rápido posible

El TCP establece una conexión vía un método conocido como saludo de tres via:

- El host requiriente envía un segmento con una opción SYN colocada en el campo CODE BITS conteniendo un número de secuencia inicial.
- El host receptor envía su número secuencial inicial de vuelta al host que hizo la petición, de nuevo en un segmento con la opción SYN establecida. También aprovecha la oportunidad de colocar un ACK del número de secuencia inicial en el segmento para el host que hizo la petición.
- El host requiriente ahora recibe el ACK del número de secuencia inicial asi que sabe que el otro lado está sincronizado con el. Todo lo que resta es que el host haciendo la petición confirme que ha recibido el número de secuencia inicial del host que recibe.

Normalmente, los segmentos son pasados por la saludo/handshake no contienen data. Note que la confirmación enviada son para el número de secuencia más uno. La confirmación apuntan al próximo byte de data que puede ser enviado.

Cualquier de los dos puntos pueden cerrar la conexión. Un segmento de FIN significa que la aplicación en la otra punta no estará enviando más data. Cada FIN deberá ser confirmado para que el que lo envió sepa que fué recibido.

Puede que piense que el ACK del primer FIN se aproveche y se envíe con el segundo. Pero, el segundo FIN sólo se generará una vez la aplicación remota cierre sus puntos de envío, asi significando que no enviará más data. Aunque un punto haya sometido un FIN, el otro punto puede continuar enviando data infinitamente. De hecho, algunas aplicaciones hacen uso de este tipo de conexiones media cerradas/half-closed. Solamente cuando ambos lados han sometido su confirmaciones a FIN es que la conexión será completamente cerrada.

Las siglas mss significan Maximum Segment Size (Tamaño máximo de Segmento) y es un limite en el tamaño máximo de las tramas de data que pueden ser enviadas al host. Esto es primordialmente útil si el host esta en una red con un MTU reducido y desea evitar demasiada fragmentación de la data.

La razón para tener cuatro señales de terminación pero sólo tres para establecer la conexión es que cada lado de la conexión debe ser apagado individualmente. El TCP es de comunicación full-duplex y se puede lograr half duplex ejecutando un half close, e.j., apagando un lado de la conexión.

Opciones TCP

Un paquete TCP puede tener opciones. Uno de los parámetros pasados por estas opciones es el Tamaño máximo de Segmento (Maximum Segment Size). Otra es la opción el Sello (Timestamp) la cual proveé una manera de registrar el tiempo que toma el viaja de ida y vuelta de la data TCP.

Control de Flujo TCP

La conversación entre el emisor y el receptor siempre incluyen una indicación de cuantos bytes de data pueden ser recibida a la vez sin riesgo de perdida, esto es llamado una ventana/window. El tamaño de la ventana es el número máximo de bytes que pueden ser enviados sin recibir una confirmación ACK. Cada segmento TCP contiene un valor de la ventana actual, en bytes, de la otra punta de la conexión.

Cuando la data es enviada, esto tiene el efecto de cerrar la ventana, en efecto, reduciendo el número de bytes que puede ser enviado actualmente. Cuando el otro lado del conexión recibe data, enviará una confirmación del número de bytes que ha recibido. Cuando se recibe la confirmación, esta tiene el efecto de abrir la ventana, en efecto, incrementar el número de bytes que pueden en la actualidad ser enviado. Este tipo de control de flujo es conocido como sliding window ya que la ventana se desliza sobre los bytes enviados.

El emisor puede calcular la ventana actualmente disponible del receptor del tamaño anunciado y del número de bytes enviado pero no confirmado. Cuando esta se convierte muy pequeña, el emisor no transmitirá hasta que el receptor anuncie que más data puede ser recibida.

Por ejemplo, si el receptor ha estado anunciando una ventana de 4,096 bytes en el segmento de su cabecera y el emisor ha enviado 2,000 bytes sin haber recibido una confirmación, la ventana restante fuese de 2,096 bytes. Si el emisor fuese a enviar otro de 2,096 bytes, el cual también es confirmado, este tendrá que restringirse de envíos posteriores hasta que no se reciba una confirmación. Si este entonces recibe una confirmación de los primeros 800 bytes enviados, entonces este podrá enviar otros 800 bytes.

El receptor almacenará los segmentos sin importar el orden en que fueron recibidos. Si el segmento llega fuera de orden o con faltante, el receptor confirmará el número de secuencia más grande, en efecto, el último punto en el flujo de data que este ha recibido la totalidad de los bytes. Este entonces esperará que el emisor de la señal de tiempo vencido (time out) y reenviará cualquier segmento que cree sea perdido.

Si se recibe un segmento duplicado, será descartado. Si se recibe un segmento fuera de orden, el último confirmado será reenviado. Asi que las confirmaciones duplicadas pueden darle al emisor una indicación de cual fué el primer segmento perdido.

SYN Cookies

Los SYN cookies son un mecanismo importante de seguridad en contra de ataques TCP SYN efectuados sobre un sistema. En la eventualidad de un ataque, SYN cookies vencen el ataque conservando memoria y no permitiendo que multiple peticiones de paquetes agoten la memoria. En esta sección discutimos como un desbordamiento (flooding) de TCP SYN funciona y como los SYN cookies puede ser usado para evitar caídas del sistemas a traves de ataques SYN.

SYN Flooding

Es muy posible que un servidor TCP/IP Web se tornará lento y hasta puede fracasar completamente si demasiadas peticiones de conexión se reciben simultáneamente. Este problema, bajo circunstancias legitimas, puede ser un problema de conexiones lenta o de un servidor que no puede procesar la alta demanda de paquetes TCP de conexión. Muy común es que personas (crackers) intencionalmente queriendo causar daños ataquen un servidor Web con un simple proceso conocido como SYN flooding, hoy día más comúnmente llamado Denial of Service attack o DoS.

Los ataques SYN flooding se aprovechan de la secuencia de eventos requeridas para establecer una conexión TCP. Normalmente, cuando una máquina se conecta a un servidor, esta envía un paquete TCP conocido como un paquete SYN (synchronization), cual el servidor Web recibe y responde enviandole un paquete de confirmación (ACK). Una ves la máquina conectandose recibe el paquete ACK, esta envía otro paquete TCP

con la opción ACK establecida, asi completando el proceso de conexión. Este proceso es a veces referido como un saludo de tres vía. Pero, cuando estamos bajo un ataque SYN flood, un gran número (posiblemente millones) de paquetes SYN son enviados, todos reclamando haber provenido desde equipos no existentes, a traves de creando fantasmas (spoofing) de direcciones de redes. El servidor trata de completar el saludo de tres vía, pero ya que la petición es de un sistema no existente, no pasa nada y el servidor deja el intento de hacer la conexión.

Un atacante usando un SYN flood puede fácilmente sobrecargar la memoria ya que cada petición de conexión requiriendo información del servidor afecta nuestro sistema. Los servidores Web son limitados en el número de paquetes SYN que ellos pueden recibir por puerto. Cuando paquetes no procesados empiezan a sobrecargar un puerto, los siguientes paquetes que arriban son descartados. Durante un ataque SYN flood, estos paquetes descartados con mucha seguridad que también eran parte de los paquetes SYN intentando conectarse o peor aún, si es un sistema intentando conectarse.

La Defensa SYN Cookie

Una defensa incluida en el kernel de Linux desde la versión 2.0.30 en adelante es encontrado del desborde SYN flooding es usar los SYN cookies (también conocido como los TCP SYN cookies). Los Cookies pueden diferenciar entre las conexiones legitimas y las ilegitimas sin usar muchos recursos de memoria. Los SYN cookies permiten que sistemas legitimo se conecten durante un ataque de SYN flood que normalmente fueran rechazados. Los SYN cookies trabajan codificando toda la información acerca de una conexión en el paquete ACK que es retornado al sistema que se conecta después de que el paquete SYN inicial es recibido por el servidor. Si la conexión es un actualmente un emisor y no desde una dirección IP falsa, el SYN cookie retornará el tercer paquete TCP que contiene toda la información necesaria sobre el sistema que se conecta. Esto significa que nada tiene que ser almacenado en la memoria que por lo normal bloquearía el servidor durante un desbordamiento SYN.

Para habilitar los SYN cookies en GNU/Linux, ejecute los siguientes comandos (como el root):

echo 1 > /proc/sys/net/ipv4/tcp_syncookies

para deshabilitarlo, sólo repita la sentencia anterior pero asigne un cero 0 en vez de un valor de 1;

echo 0 > /proc/sys/net/ipv4/tcp_syncookies

Puertos y Sockets

Necesitamos un método para canalizar los datos que fluyen desde y hasta a un proceso en particular de una aplicación. Este meto de canalizar datos es conocido como puertos y sockets. Un socket es un par de pila de datos para el proceso de una aplicación, uno en el punto de enviar o input y uno en el punto de recepción o el output. Un puerto es la dirección de un socket, igual al número de la extensión telefónica es la dirección de esa extensión.

Un Sockets por general es implementado como un área de memoria dentro del kernel de un sistema operativo. La data entrante es es destinada vía la dirección del puerto de un socket y por lo tanto directamente dentro del proceso de una aplicación.

El protocolo de transporte usa el concepto de un puerto para representar un punto d comunicación. Los puertos son asignados números, representados por un valor de 16-bit. Esto significa que ellos pueden ser hasta 64 KB de puertos individuales. El TCP y UDP cada uno tiene su propio conjunto de puertos.

Los puertos usados usados por TCP/IP son parecidos a los números de extension de un sistema telefónicos. Si deseamos contactar a un amigo via teléfono debes primero tener su número telefónico. Esto es el equi-

valente a usar una dirección IP para contactar un host. Una vez y lo has contactado, puede que necesite pedir una extensión particular para poder hablar con su amigo. Este es el equivalente al número de puerto.

La combinación de dirección IP y el número´mero de puerto es a menudo mejor conocido como un socket. Al combinarlo con el número de protocolo seleccionado, ya sea TCP o UDP, esto provee una dirección única de u proceso en un red basada en TCP/IP. Esto en en efecto es, un par de colas para la data de la aplicación, una de entrada y otra de salida.

El nombre del socket también es referido como un API (Application Programming Interfase), usado frecuentemente para el desarrollo de aplicaciones TCP/IP. La interfase de programación de sockets fué desarrollada en la Universidad de California Berkley cuando ellos implementaban TCP /IP en UNiX. Por esto a veces es referido como los sockets de Berkeley.

La interfase de sockets soporta toda la funcionalidad que un programador necesita para implementar un cliente o un servidor. Esta es una interfase por defecto en UNiX y ahora es común en otros ambientes que soportan sockets TCP/IP.

Número de Puertos

Los servidores escuchan por peticiones de clientes en puertos específicos, como esta definido en el RFC 1700. Los puertos números 1 al 1,023 son manejados por IANA. Teniendo los procesos de servidor escuchar por comunicaciones en u punto en particular es como los servicios son anunciados. Es normal tener los procesos del servidor escuchar en el todos los sistemas que proveen el servicio. Este es a menudo llamado el puerto de reencuentro. Por ejemplo, para contactar el servicio de telnet en un sistema, tendría que comunicarse usando TCP, con el puerto número 23 en ese sistema.

Los números de puerto entre 0 y 1,023 son reservados para servicios bien conocidos y solo pueden ser accesados por root. Estos puertos también son referidos como puertos privilegiados o o servicios bien conocidos. La mayoría de los servicios que se ejecutan en el Internet utilizan estos puertos bien conocidos. Esto es para que estos servicios siempre puedan ser contactados en el mismo puerto sin importar la dirección IP del sistema. Los usuarios que no son root pueden usar los número s de puertos que son de número más alto que el 1,024. Por eso es que puede ser vea el servidor HTTP ejecutándose en el puerto 8,000 o el 8080- ellos son ejecutados por otros usuarios que no son root ni otros administradores del sistemas.

Los servicios bien conocidos son definidos por una organización central de nombre IANA. Aunque los número de puertos TCP y UDP son independiente, Típicamente son asignados el mismo número bien conocido para ambos el TCP y el UDP.

Los procesos cliente requieren diferentes puntos para la comunicación, al igual que los servidores también. Eso si, no es importante que un cliente use siempre el mismo número de puerto. Cualquier número de puerto puede ser usado siempre y cuando el número de puerto usado es consistente durante una conversación con el servidor. Para permitirlo, TCP y UDP pueden dinámicamente seleccionar un puerto que no este en uso para (por lo general con números muy alto) darle a un cliente que lo requiere. Estos puertos son conocidos como números de puertos efímeros.

Los servidores escuchan por peticiones de conexiones en estos puertos específicos porque los clientes conocen estos puertos, y porque los puertos están estandarizado.

Números de Puertos Importantes

Algunos números de puertos han sido reservados para servicios comunes servidos en todo el Internet. Los listamos en la siguiente tabla.

Puerto	Servicio
7	Echo
13	Date + Time
20&21	FTP
22	SSH
23	Telnet
25	Mail (SNTP)
57	Domain Name System (DNS)
80	www (HTTP)
110	Mail (POP3)
111	Portmapper (RPC)
119	News(NNTP)
123	Network Time Protocol (NTP)
137	NetBIOS name service (WINS)
139	NetBIOS session servile
143	Internet Message Access Protocol (IMAP2)
161,162	Simple Network Management Protocol (SNMP)
389	lightweight Directory Access Protocol (LDAP)
443	Secure Sockets Layer (SSL)

Usted puede ver que números de puerto están disponible para brindar servicios en su sistema si damos lectura al archivo /etc/services. Este archivo contiene el mapeo de los nombres de servicios a los números de puertos. Algunos programas nos permiten usar el nombre del servicio en vez del número de puerto. Por ejemplo, ambos de los siguientes comandos abriran una conexión al servidor Web en la máquina de nombre web.abiertos.org:

```
telnet   web.abiertos.org 80
telnet  web.abiertos.org  www
```

Pasar Data a las Aplicaciones

Cuando un datagrama IP es recibido por un host, este puede determinar cual protocolo de transporte este debe ser entregado examinando el valor en el campo protocolo del cabezal IP. Cada uno de los protocolos de transporte, TCP y UDP, soportan una amplia gama de servicios al nivel de aplicación. Es perfectamente posible, al punto hasta generalizado, tener más de un servicio activo a la vez. Asi que debemos tener una manera de identificar comunicaciones individuales en ambos puntos desde dentro del software del protocolo de transporte.

Como ya hemos visto, ambos cabezales del TCP y UDP contienen dos números de puertos, uno de destino y otro de origen. El número de puerto de destino determina a cual aplicación la data se le pasa.

Ahora un programador de aplicaciones solo tiene que preocuparse de el flujo de data que el esta usando y puede dejar que el TCP maneje la transferencia confiable, de flujo controlado. Esto simplifica enormemente la tarea de escribir aplicaciones.

El Protocolo HTTP

El protocolo HTTP (Hypertext Transfer Protocol) esta definido en el RFC 1945 como el del World Wide Web. El tráfico WWW es pasado por el puerto 80, cual es el HTTP, asignando por el RFC 2086 en su versión 1.1. El SSl (Secure Sockets Layer) usa el puerto 443, conocido como el HTTPS.

Es importante denotar que el HTTP 1.0 requiere conexiones por separado para cada descarga de archivo o imagen. El HTTP 1.1 si permite que las conexiones se mantengan abiertas.

Recientemente desarrollado en el año 1992 por el científico Británico Tim Berners-Lee, el HTTP ha evolucionado para proveer un mecanismo estandar por el cual documentos pueden ser publicados y presentados en línea.

El protocolo HTTP es usado para transferir páginas WWW. Las páginas son escritas en el lenguaje de marcado de hipertexto HTML (Hypertext Markup Language). El HTML permite la especificación de diagramación, fuentes, colores y inclusión de gráficas en una página WWW. El HTML también define vínculos, los cuáles pueden ser usados para navegar de página en página a traves de efectuar clicks.

La aplicación que provee el servicio de vaginas y documentos WWW es llamada un servidor de páginas Web. El Apache (www.apache.org) es el servidor de páginas web más popular en uso en el Internet con una distribución de uso de más de un 70% y creciendo. La aplicacion cliente que utiliza los servicios WWW es llamado un navegador/browser, y presenta el material gráficamente y con el formato deseado en pantalla al usuario.

El SSL proveé un flujo de data encriptado que es utilizado para proveer comunicaciones encriptadas y seguras en el puerto 443 y el protocolo seguro HTTPS (Secure HTTP). Los sitios web que ofrecen servidores web con el SSl habilitado sobre el puerto 443 tienen URLs que se digitan con “https://dirección.dominio”.

Servidores GNU/Linux

GNU/Linux ofrece una gran variedad de servidores para los diferentes protocolos de alto nivel. Estos son identificados por el nombre común del protocolo más una “d”, e.j., telnetd y sshd. Los servidores también son conocidos como daemons en el léxico de GNU/Linux.

Los servidores o daemons TCP tienden a ser concurrentes; lo que significa que ellos esperan por la petición de un cliente, inician el nuevo proceso/tarea/hilo para manejar el cliente, luego retornan a esperar por la próxima petición.

En implementaciones anteriores de TCP/IP en GNU/Linux, un servidor tenía que estar ejecutándose por cada servicio que se ofrecía en su host. Pero, la mayoría de las veces, mucho de estos servers estarían inactivos, esperando por clientes. Lo cual significa que muchos recursos del sistema están siendo desperdiciados. Implementaciones más recientes usan una tecnología llamada los super server, su nombre en GNU/Linux es inetd, el cual espera por conexiones en todos los puertos y entonces enciende/spawns el servidor de la aplicación cuando la petición llega. Lo cual significa que sólo el super server se encuentra ejecutándose a menos que alguien este actualmente usando el servicio. El super server es frecuentemente usado y es configurado vía el comando /etc/inetd.conf. En GNU/Linux, el TCP/IP es altamente optimizado, asi resultando en implementaciones mucho más rápido de TCP/IP.

Ejercicio 4-1: Seguridad de la Red de la Compañía X

No se proveen soluciones a este ejercicio. La red está ya en un estado muy avanzado. Usted es el administrador de la red. Aqui se le provee las necesidades de diferentes departamentos de la Compañía X, y usted está en el deber de proveer estas necesidades pero siempre tomando en cuenta la necesidades de seguridad.

Es necesario que piense en todo el disenso de la Compañía X. Trate de responder las siguientes preguntas:

1.- ¿Cuáles medidas de seguridad de la red puede usted usar?

2.- La oficina principal no esta expuesta al Internet. ¿Qué tipos de defensas son apropiadas para estos siste-

mas?

3.- El departamento de Ingeniería necesita la máxima protección. ¿Qué sugiere usted aquí en este caso?

4.- El equipo de desarrollo necesita acceso rápido y fácil a recursos en línea pero posee datos sensibles que no pueden ser robados. ¿Puede usted pensar en una solución que satisfaga ambas de estas necesidades?

5.- También considere que otras sugerencias y medidas de seguridad puede usted tomar para asegurar que nuestra oficina no sea blanco de ataque.

Ejercicio 4-2: Completar la Configuración de Red

No se proveen soluciones a este ejercicio. Deberá tener los permisos de superusuario para editar el archivo hosts. Si lo edita en vi y le informa que está en modo de solo lectura, solo tendría que al momento de guarda hacerlo con el comando :w! de vi para forzar vi a escribir el archivo.

Este es un ejemplo del archivo hosts:

```
# Mi archivo hosts
127.0.0.1       localhost
192.168.1.5     almacen
192.168.2.7     proxy
```

Ahora usted debe poder hacer ping a estos equipos solo usando el nombre y no el IP. Aseguro que usted puede hacer esto.

Pruebe haciendo ping a un sistema en otra red (use la dirección IP del sistema). ¿Qué mensaje usted ve?

Ahora asignele a uno o más de los hosts en su archivo hosts un alias. Revise que ahora pueda contactar el sistema usando:

- **Dirección IP**
- **Nombre de Host**
- **Alias**

Acceso/Login Remoto a Otros Sistemas

Usted debe poder poder repetir su login remoto a otros sistemas, pero esta vez usando el nombre del sistema y no la dirección IP.
Use el comando telnet para accesar el sistema remoto:

$ telnet nombre-sistema
Este comando le presentará un prompt pidiendole un nombre de usuario y contraseña.

Una vez usted a ingresado a sistema remoto, escriba el siguiente comando:
$ uname -n
Este comando imprimirá el nombre de host del sistema que usted se encuentra, logueado remotamente.

Monitorear la Actividad de la Red

El comando netstat es útil ya que nos da la habilidad de monitorear el estatus del software utilizando la red en el sistema. Para esta sección, usted debe disponer de por los menos dos ventas abiertas; una para ejecutar el comando netstat para examinar su salida y la otra para ejecutar comandos de red.

Escriba el siguiente comando:
$ netstat -a --inet (netstat -af inet en algunos sistemas)

Para desplegar el estado de los puntos TCP y UDP en su sistema.

Al examinar los punto del TCP en su sistema, usted debe ver que las mayorías de puntos están en un estado de LISTEN/Escuchando. Esto significa que no haya conexiones activas en este momento, pero que el servidor esta activo y escuchando.

Ahora desde la otra venta que abrimos, ingrese en un sistema remoto usando el comando telnet, como hizo anteriormente. Una vez ingresado al sistema, escriba el siguiente comando:

$ netstat -a --inet

Note que ahora su conexión ya esta indicada. La sesión de telnet mostrará un esta de conexión en el estado de de ESTABLISHED/establecida. Si otros usuarios se encuentran ingresados en su sistema, sus conexiones también estarán marcadas. Usted debe poder deducir cuáles sistemas tienen usuarios logueados en la actualidad en su sistema desde esta información.

Si usted escribe:

$ netstat -ei

Usted será presentado un resumen de la estadística de las interfases de redes conectadas e instaladas al equipo. Alternativamente, el comando netstat puede ser ejecutado con la opción -c para continuamente actualizar su salida (aunque esto tiende a ser difícil de leer, particularmente si la salida no es alimentada por tubería a un paginador o lector). Pruebe con la combinación de opción -ae para ver que conexiones son para que servicios.

Cada servicio basado en TCP o UDP en su sistema esta asociado con el número de un puerto. Los detalles de los servicios, sus protocolos de transporte y los números de puertos se almacenan en el archivo /etc/services.

Usando sólo la información en este archivo, llene el siguiente formulario de los protocolos de transporte y los números de puertos usados por los siguientes servicios:

time:	**---------------**	**---------------**
echo:	**---------------**	**---------------**
tftp:	**---------------**	**---------------**
name server:	**---------------**	**---------------**
telnet:	**---------------**	**---------------**

Ejecute de nuevo el comando netstat con la opción numérica y observe como esto se refleja con el número services:

$ netstat -an --inet

Finalmente usted debe estar consciente como es que GNU/Linux controla los servicios principales. Esto se resumen en las siguientes secciones. Read the following information to raise your awareness.

El inetd

GNU/Linux, al igual que su antecesor UNiX, posee mecanismos de ahorro de memoria para proveer servicios vía el proceso de super-server. El mapeo de los puertos a los servidores tiene dos etapas:

- El archivo /etc/services mapea los número de puertos al nombre de los servicios.
- El archivo /etc/inetd.conf mapea los servicios a los nombres de los servidores.

El inetd monitorea todos los puertos en el archivo inetd.conf esperando por intentos a conexiones. Al recibir una en un puerto en particular, establece la asociación con el nombre de servicio asociado a este puerto en el archivo services. Ya establecido esto, busca en la información que recibió del archivo inetd.conf. Esto le dice varias cosas acerca de la conexión, la más importante siendo donde reside el servidor de ese servicio, entonces inetd iniciará el servidor y entregará la conexión al servidor. Luego retorna a escuchar en el puerto. Las conexiones individuales al mismo servicio o número de puerto son diferenciadas por la combinación de la dirección única de IP y el número de puerto.

El archivo de servicio es un mecanismo extremadamente flexible. Con simplemente cambiando la entra-

da relevante, es posible alterar el puerto que el servicio usará de ahí en lo adelante.

Este es un formato básico del archivo inetd.conf:

Servicio	Endpoint	Protocolo	Wait	Usuario	Ruta	Argumentos
finger	stream	tcp	nowait	nobody	/usr/libexec/tcpd	fingerd-s
ftp	stream	tcp	nowait	root	/usr/libexec/tcpd	ftpd-l
login	stream	tcp	nowait	root	/usr/libexec/tcpd	rlogind
nntp	stream	tcp	nowait	usenet	/usr/libexec/tcpd	nntpd
ntalk	dgram	udp	wait	root	/usr/libexec/tcpd	ntalkd
shell	stream	tcp	nowait	root	/usr/libexec/tcpd	rshd
telnet	stream	tcp	nowait	root	/usr/libexec/tcpd	telnetd
uucpd	stream	tcp	nowait	root	/usr/libexec/tcpd	uucpd
comsat	dgram	udp	wait	root	/usr/libexec/tcpd	comsat
tftp	dgram	udp	wait	nobody	/usr/libexec/tcpd	tftpd /private/tftp
bootbootps	dgram	udp	wait	root	/usr/libexec/tcpd	bootpd
pop3	stream	tcp	nowait	root	/usr/libexec/tcpd	/usr/local/libexec/popper
imap4	stream	tcp	nowait	root	/usr/libexec/tcpd	/usr/local/libexec/imapd

El super servidor inetd, /usr/shin/inetd, lee la información que describe los servicios que el ha de controlar desde el archivo de configuración /etc/inetd.conf. Las entradas en este archivo describen los servicios de la siguiente manera:

Service Nombre del servicio. Debe corresponder con una entrada en /etc/services y es usado para decirle a inetd cual puerto monitorear para este servicio.
Endpoint Define si el punto final del servicio va a operar las transferencias con flujos/stream o basados en datagramas.
Ptcl Este es el protocolo que es usado para el servicio.
Wait Define si múltiples clientes concurrentes son soportados (nowait) o no.
User El nombre de usuario bajo el cual se ejecuta el servicio.
Pathname La ruta del programa a ejecutarse.
Arguments Los argumentos (incluyendo el nombre del comando) pasados en el momento de e al programa servidor.

Para desactivar un servicio simplemente se elimina la entrada del archivo /etc/inetd.conf o se coloca (#) como un comentario al inicio de la línea que queremos anular. El archivo es leído una vez cuando el inetd se inicia al momento de boot, los cambios deben ser notificados a inetd. Esto se logra enviandole una señal de SIGHUP (señal no. 1), por ejemplo:

```
# ps  -aux  |  grep inetd
root  321    0.0     1.2      892      392     ?    S    May  13     0:22 inetd
#  kill   -HUP 321
```

El Protocolo UDP

El TCP si es complejo; este debe proveer confiabilidad (confirmaciones/acknowledgments), flujo (saludo/handshaking), control de flujo (tamaño de ventana/windowing), soporte para interrumpir el flujo y más. Todo esto suma a la complejidad del TCP y subtraer del desempeño del protocolo. Para algunas aplicaciones, estos requerimientos no son necesarios. Estas aplicaciones necesitan una alternativa, UDP.

Los siguientes tópicos son discutidos en esta sección:
• Fundamentos de UDP
• Transferir Archivoa vía TFTP
• Números de Puertos UDP

Fundamentos de UDP

El UDP puede ser usado cuando confiabilidad no es importante y cuando no es necesario el flujo de data, esto significa que no hace falta la numeración de las tramas.

Formato de la Trama

El cabezal incluido en cada datagrama por el UDP es simple, consiste principalmente de información que permite que el UDP efectué su entrega de la data al proceso apropiado (los números de puerto).

El cabezal UDP contiene los siguientes campos.

- **Source Port (16 bits)**
  Campo que define el número de puerto del servicio que genera el mensaje en nodo de origen.
- **Destination Port (16 bits)**
  Este campo identifica el número de puerto del servicio en el nodo destino.
- **Message Length (16 bits)**
  Este campo identifica el número de bytes en el paquete UDP (incluye el cabezal UDP y la data). Esto es realmente redundante ya que el cabezal UDP es de longitud fija y el tamaño de la data UDP puede ser calculada tomando en cuenta solamente la longitud del datagrama IP. La longitud es un valor sin signo de 16-bit, dando un datagrama UDP máximo de 65,535 bytes, o 64 KB.

- **Checksum (16 bits)**
  Este campo es opcional. Un checksum de valor 0 implica que el checksum no fué corrompidos. El Checksum cubre toda la data en el datagrama UDP, su cabezal de 8-byte y también la dirección IP de origen y destino y el valor del protocolo de la dirección IP. Si el receptor detecta que el datagrama esta corrompido, simplemente lo descartas, al igual que el IP. Es el deber de las aplicaciones de más alto nivel usando el UDP efectuar su propia detección y corrección de error.
  Algunas aplicaciones deshabilitan el checksum del UDP para reducir el costo del uso de recursos. En una LAN, esto es razonable ya que la capa de Enlace por lo general detecta cualquier tipo de corrupción. Pero, al trabajar sobre el Internet (internetwork), el checksum debe siempre estar habilitado ya que no existe otra manera para el receptor revisar la integridad de la data.

El cabezal combinado con la data es encapsulada como la porción de data del datagrama de IP. El datagrama IP también contiene las direcciones del sistema de origen y destino y el campo del Protocolo IP contendrá un valor de 17, indicando que la data es un datagrama UDP.

Cuestión de Confiabilidad

Claro está, este protocolo no es útil cuando es importante la confiabilidad. Aquí le presentamos una lista de servicios que por lo general dependen de una conexión confiable y debido a esto no son recomendable a través del UDP:

- FTP
- Mail
- Telnet
- WWW

El UDP en la Practica

El UDP es un protocolo simple, que esencialmente provee una interfase de alto nivel a la facilidades del IP para transportar datagramas. La data es transmitida en datagramas sin garantía de entrega; si confiabilidad es importante, entonces es delegada al software de más alto nivel para que la provea.

El UDP coloca un bajo costo fijo en termino del control de información de protocolo en la data transmitida. Es muy a menudo considerado mucho más eficiente que su alternativa el TCP, y realmente en un ambiente LAN, por lo general es más que adecuado como un protocolo de transporte. Muchas aplicaciones de área local utilizan el UDP por esta razón. En redes de gran escala que involucran muchos gateas y enlaces de mayor área, la falta de un mecanismo confiable de entrega interno del protocolo se convierte en un problema y en este caso el TCP se convierte en una mejor elección.

Generalmente los usuarios accesarla red a través de un servicio del nivel de aplicación, y la elección entre el UDP y el TCP ya habrá sido efectuada por el programador de la aplicación.

En la manera que un servidor se comporta depende de que protocolo de transporte este usa. Los servidores que usan UDP por lo general serán iterativos. Lo que significa que ellos manejaran cada petición a medi-

da que lleguen y no atenderán otras peticiones hasta que no terminen con la actual. Esto cabe en el hecho que los datagramas UDP son entidades independiente; asi que ellos deben contener toda la data necesaria para identificar la petición.

Transferencia de Archivos vía TFTP

El TFTP (Trivial File Transfer Protocol) es asignado el puerto UDP número 69 definido en el RFC 783. El TFTP es mucho más pequeño y menos sofisticado que el FTP. Este usa el UDP como su medio de transport, lo cual simplifica considerablemente los programas clientes y servidores. También contiene un conjunto mucho más pequeño de comandos que solo se conciernen con la transferencia de archivos.

En una sesión básica empieza con una petición de escritura desde el cliente al puerto UDP 69 del servidor. El servidor entonces responde con una confirmación. Pero note que que esta confirmación y las conversaciones de ahi en lo adelante usaran un puerto en el servidor UDP diferente, el puerto es el 1390, el cual es un puerto efimero/ephemeral port. Las conversaciones continúan y concluyen en este puerto y en el puerto original del cliente.

¿Por qué es esto? Bien, el server TFTP desea liberar el puerto 69 lo más pronto posible. Bajo el TCP, tenemos un protocolo orientado a conexión con una conexión identificadas por ambos puntos de la conversación, como es, la dirección IP del cliente y el servidor, los puertos en cada punto y el protocolo (TCP). Asi pues, es perfectamente posible tener múltiples clientes conectarse al mismo puerto TCP del servidor porque la combinación de IP y puerto del cliente serán diferente para cada conexión.

Pero bajo el UDP, no tenemos una noción de conexiones. Es perfectamente posible que el servidor TFTP mantenga información de los IPs y puertos de los clientes y asi poder manejar multiple conexiones, pero todo esto tiene que ser llevado acabo en al nivel de las aplicaciones y requieren un esfuerzo adicional del programador. Además, lo más seguro que disminuiría la velocidad del servidor TFTP, y la intensión del TFTP es proveer un método simple y rápido de transferir archivos, y esto agrega una complejidad innecesaria. Asi pues, el el caso previo, el servidor TFTP resuelve este problema simplemente capturando un puerto efimero número 1390 y lo usa, y se va del puerto 69.

Si buscamos en GNU/Linux en el archivo /etc/inetd.conf, verá que el tftp esta marcado con "wait"; esto es que, las nuevas peticiones TFTP al puerto UDP 69 deberán esperar hasta que se libere para aceptar nuevas peticiones.

Cada transferencia empieza con un cliente efectuando una petición de lectura (RRQ) o de escritura (WRQ) al servidor. Una vez el servidor se ha asegurado que puede leer o escribir a un archivo dado, el host enviando la data envía el primer bloque. El receptor confirma el block antes de que otros bloques más sean enviados. Si no se reciben confirmaciones dentro de un tiempo específico, el emisor reenviará el bloque. El TFTP es un ejemplo de un protocolo pare-y-espere (stop-and-wait).

Debido a su pequeño tamaño y simplicidad, el TFTP es adecuado para la transferencia de archivos de inicialización durante procedimientos de arranque/boot. Es posible codificar un cliente TFTP para que quepa en un ROM en máquinas sin discos/diskless y asi formar parte del bootstrap (conjunto de archivos encargado de dar inicio al equipo) para esa máquina.

El TFTP no autoriza los usuarios como lo hace el FTP. Esto es un potencial punto débil de seguridad, ya que cualquiera puede tener acceso a copiar archivos desde y a nuestro sistema.

Números de Puertos UDP

Los servidores escuchan o atienden puertos específicos mientras esperan por las peticiones de los clientes, como es ilustrado en el RFC 1700. Los números de puertos del 1 al 1,023 son administrados por IANA, y son llamados los puertos bien conocidos o well known services. El rango entre el 1,024 al 5,000 son reservados para los procesos de los clientes, clientes son las aplicaciones. Los puertos por encima del 5,000 son conocidos como los NO bien conocidos (Non-Well known). Son utilizados para las aplicaciones de redes desarrolladas internamente en una empresa.

UDP tiene su propio conjunto de número de puertos; estos se ajustan exactamente igual que el esquema de los números de puertos TCP. Cuando el IANA asigna un número de puerto a un servicio, le asigna este número de puerto a ambos el TCP y el UDP, aunque el servicio no los use a ambos.

RESUMEN

En este capítulo, estudiamos el ICMP y los protocolos de la capa de transporte, UDP y TCP. Los siguiente es un lista de los puntos más importantes a recordar:

- ICMP es un protocolo simple usado para ayudar a mantener el Internet.
- ICMP es poderoso y puede rápidamente detectar donde empiezan y ocurren problemas de red.
- UDP es usado por servicios sin conexiones donde confiabilidad no es importante o necesaria.
- TCP es usado para servicios orientados a conexión y donde se necesita cierto grado de confiabilidad.
- TCP es más complejo y conlleva ciertos costos fijos en sus servicios de transporte.

PREGUNTAS POST-EXAMEN

Respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué ventajas tiene el TCP sobre el UDP? ¿Cual es usado más a menudo?

2. Describa el Cabezal IP.



RESOLUCIÓN DE AVERIAS Y ADMINISTRACIÓN DE RED

OBJETIVOS

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Dar solución a problemas de averias de TCP/IP
- Describir tecnicas comunes en la solución de problemas de redes.
- Comparar y contrasta las herramientas gráficas de configuración de redes: YaSt2, Netconf, NetConfig, y Turbonetconf,
- Describir la herramienta de administración de la red Netperf, particularmente los servicios que el programa proporciona.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son los tres pasos dado para comprobar para saber si hay conectividad?

2. ¿Durante el uso de un ftp, como podía corrumpirse un archivo?

3. ¿Qué efectua el netstat?

4. ¿Cuáles son los tres componentes de un sistema moderno de la dirección de la red?

INTRODUCCIÓN

Muchas herramientas están disponibles para las redes en la localizacion TCP/IP. Este capítulo introducirá los archivos, las utilidades, y los comandos más importantes para localizar averías y discutirá cómo utilizarlos. Porque éstos tomaron se utilizan comúnmente con las redes de TCP/IP, muchas de ellas pueden ser modificadas y ser utilizadas a través de cualquier plataforma que funciona la habitación de TCP/IP.

La resolución de problemas del funcionamiento de una red de TCP/IP en el Distribución anbiente de sistema de hoy requiere analizar el efecto de la actividad del cliente y del servidor en sus sistemas, la red, y sus usos client/server.

Típicamente, el usuario del extremo es el primer para notar si una red se está realizando mal.

Este capítulo identifica los elementos que afectan el funcionamiento de las redes de TCP/IP. También cubriremos cómo los parámetros se asociaron a sus sistemas, la red, y sus usos client/server se pueden optimizar para reducir al mínimo problemas de la red. Los ejercicios de la resolución de problemas a través del capítulo asociarán los conceptos dominantes a problemas de funcionamiento reales de la red.

Mientras que i: no es posible cubrir una tesis profundizada 011 la resolución de problemas, este capítulo le señala en la dirección derecha.Con estas extremidades, usted puede buscar los libros, RFCs, cuartos de la charla, y newsgroup de ayudar a encontrar el ro de las soluciones sus problemas del establecimiento de una red. Este capítulo también da extremidades de la dirección de la red.

FUNDAMENTOS DE RESOLUCION DE PROBLEMAS

En esta sección, hecharemos una ojeada centi'metro-sory cómo TCP/IP que localiza averías se logra. examinamos los varios métodos para localizar averías y las herramientas y los procedimientos que están disponibles para asistirle.

En esta sección discutimos los siguientes temas:
- Principios generales para localizar averías
- Caja de herramientas De Troubleshooter.
- Problemas Físicos más Comunes
- Problemas Lógicos más Comunes
- Problemas Especisicacion de Protocolos

Principios Generales para Localizar Averías

Hay muchas metodologías para localizar averías. Ni continuar una discusión exhaustiva de metodologías; algo, hecharemos una ojeada una estructura básica de la metodología para localizar averías, aplicable en la mayoría de las situaciones.

Reconozca los síntomas

Muy a menudo, uno no puede solucionar el problema porque uno no lo ha identificarlo correctamente.

Investigue el problema

Debe siempre asegurarse de identificar donde y cuando el problema comenzó tan bien como cuántos sistemas pueden ser afectados semejantemente.

Adquiera Información

Este paso puede ser una consecuencia del paso anterior. Se cerciora de siempre usted tener toda la infor-

mación disponible de las fuentes, de la gente, y de los sistemas implicados.

Pruébelo

Deje su ayuda del diagnóstico de usted y sea seguro interpretando sus resultados exactamente.

No salte a las Conclusiones

A menudo el primer impulso que usted tiene puede curar el síntoma pero no el problema subyacente. Asegúrese siempre de usted haber cavado profundamente bastante para aislar la fuente del problema, la curación no justa el síntoma.

Encuentre la Avería

Este paso se podía también llamar encontrar la causa a los efectos descubiertos.

Lista de lo qué Podría Ser

No subestime la reunión de reflexión y la suerte en este paso. Ayuda a menudo a conseguir varios diversos pares de ojos y la habilidad fija implicado en identificar causas posibles (el tliat está, trabajo collabotarive de los esfuerzos mejor).

Elimine los Problemas Uno Por Uno

Cuando usted no sabe cuál es es el problema, pero usted sabe lo que podría ser, comienze a eliminar las menos probables. Usted puede encontrar a menudo que las causas son aisladas mejor con el proceso de la eliminación que intentando corregir lo que no es el problema. Recuerde que mientras más causas no posibles son eliminadas, menos el trabajo que tendrá que efectuar para corregir el problema.

Corrija la Avería

Ésta es normalmente la parte fácil. Cerciórese de que usted cure la enfermedad, no apenas el síntoma.

Analice la Falla

Ahora que no hay presión puesto que pudo corregir el problema, haga las preguntas importantes: ¿Se repetirá? ¿Es un sintoma de otros problemas? ¿Qué contribuyó a este suceso? ¿Cómo prevengo esto en el futuro?

Caja de Herramientas para Corregir Averias

Recuerde que mientras más herramientas de corregir fallas posea mejor serán sus chances de éxito. Herramientas para corregir fallas de ambas indole de software y de hardware, asi como la preparación fundamentada y la experiencia de quien le pueda ser de asistencia, pueden hacer la resolución de problemas más rápida y más fácil.

Herramientas De Software

Entre algunas de las herramientas de software útiles para asistir en la resolución de problemas TCP/IP se pueden incluir:

- ping
- netstat
- traceroute
- nslookup
- ifconfig
- lsmod
- route
- ps

Los ficheros del log pueden también ser especialmente útiles. Revise siempre y cuando sea posible los ficheros de diario, por existencia de los mensajes de error. Incluso los mensajes no relacionados con el problema particular que usted está tratando de solucionar pueden resultar provechosos asi como pueden asistir-

le a no perseguir pistas erroneas y evitar asi perder un tiempo valioso en aíslar la causa principal del problema. Siempre y cuando pueda habilite el servicio de log de mensajes, particularmente los críticos.

Herramientas De Hardware

Los analizadores de red pueden ser particularmente útiles en el diagnóstico de problemas de red, y en algunos casos, pueden hastas recomendar curso de acción para remediar los problemas. Los analizadores de red disponibles pueden ser basados en hardware o software y dedicado o nondedicados. Los fragmentos de paquete capturados (y por lo general interpretados) por los analizadores de red son el mejor método para determinarse qué realmente está sucediendo en su red. Al hacer su investigación siempre recuerde que los administradores de sistema, los usuario e incluso las máquinas y los softwares pueden mentir, pero el cable nunca mintirá.

Otras Herramientas

No se olvide de que su herramienta más útil es su cerebro y los otros cerebros que usted puede integrar a la solución del problema.

Problemas Físicos Comunes

Problemas fisíos que involucran conectividad y enrutamiento.

Conectividad

Pongale mucha atención a las cosas más simples y obvias. Muy a menudo la resolución de un problema de alto nivel puede ser simplificado observando deficiencias obvias en el sistema.

Primero, ¿Están todos los cables conectados correctamente? ¿se encuentran conectados el teclado y el ratón? ¿está el cable de red en el lugar correcto y en perfecto estado? Se han dedicado muchas horas innecesarias localizando el problema de red. Esto ocurre muchas veces, cuando sólo era un cable desconectado, por accidente u otras fallas simples. Siempre haga las cosas simples primero y busque lo obvio y no lo complejo inicialmente al empezar a localizar averías.

Enrutamiento

Si usted se encuentra en una red enrutada, lo que es la generalidad de hoy en día, la resolución de problemas en su caso llega a ser un poco más compleja para usted determinar en qué segmento o subred está la avería. La detección de esto puede implicar muchos pasos, incluyendo (pero no limitado a):

- ¿Muestra el ping pérdida de paquetes?
  - Una conexión lenta puede ser la causa de éstas pérdidas.
  - O no trabajar del todo.
- Si sus manejadores de red se cargan como módulos en el kernel, puede ser que necesite utilizar el comando lsmod para asegurarse de que el módulo apropiado este cargado en el kernel.

Luego, revise las tablas de enrutamiento para verificar que estén válidas y correctas. Utilice el comando route para comprobar sus tablas de enrutamiento IP.

Usted puede utilizar el comando netstat para visualizar el estado de las conexiones de red actuales y con la opción -a para imprimir todos los sockets, incluyen los que se encuentra escuchando por peticiones.

Asegurse de detener conexiones en ambos puntos. Los paquetes pueden fluir desde un punto pero no desde el otro, causando la confusión y enmascarándo donde sucede el problema.

Problemas Lógicos Comunes

Los problemas lógicos conciernen con archivos de configuración, un proceso del servidor, la pila del TCP/IP, servicio de nombre y archivos de diario.

Archivos de Configuración

¿Le parecen aceptables? Sea especialmente cuidadoso de los cambios recientes, especialmente ésos que coinciden sobre preguntas relacionadas con el problema en cuestión.

Procesos del Servidor

Debe preguntarse ¿Se encuentra el servidor en cuestión ejecutándose? Para determinar esto, intente conectar se con él con una sesión de telnet. Utilice el comando ps -ax para visualizar los procesos actualmente ejecutándose.

¿Es el proceso en ejecusión el proceso correcto (es decir, está el servidor en ejecusión corriendo el proceso que se supone?

No se olvide asegurarse que todos los sistemas no tengan ningún servicio extraño o innecesario ejecutándose. Éstos podrían estar interfiriendo con los servicios necesarios, y no todos los sistemas tienen todos los servicios habilitados.

La Pila TCP/IP

Primero, asegúrese que la interfaz TCP/IP está habilitada, ejecutándose y configurada correctamente. Utilice el comando ifconfig para comprobar y cerciorarse de que la interfaz está activa y con los ajustes de IP y la nermask correcta. Luego, intente de ejecutar un ping al localhost. Si esta prueba falla, usted tiene un problema de configuración.

¿Es su dirección IP correcta? Si la respuesta es no, puede ser que tenga un conflicto de dirección IP con otra máquina o que tenga una dirección inválida o corrompida.

¿Está la mascara de red (netmask) configurada correctamente? Si la respuesta es no, puede ser que se encuentre en la subred incorrecta. Esto puede conducir a muchos errores, especialmente al intentar alcanzar un host patticular.

Servicio de Nombre

¿Puede usted actualmente resolver nombres de host? ¿Puede usted resolver el nombre de host que usted está utilizando? Si la respuesta es no, intente usando la dirección IP.

Ficheros de Diario y Estadística

¿Ha generado un registro de error en los archivos log cualquier parte de la pila TCP/IP? Revise de log en el directorio /var/log/messages por cualquier reportado. Preste mucha atención en particular a estadísticas desequilibrada, e.j:, varios paquetes ICMP recibidos, ningunos enviados.

Problemas Específicos de Protocolos

Cada protocolo tiene diversos medios de la comunicación, así que cada uno tendrá un modo particular de falla. Es bueno saber algunos de los problemas comunes que pueden ocurrir en cada uno.

Problemas del Ftp

A continuación le presentamos una lista de algunos problemas comunes del ftp y cómo diagnosticarlos:

- **Tiempo de Conexión Agotado**

Recuerde que el ftp utiliza un contador de tiempo para detectar conexiones perdidas. Fijese cuánto tiempo toma un ping. Si se toma mucho tiempo, esta es la causa probable de la culpa o el factor que contribuye a esta falla.

- **Conexión Rechazada**

¿Soporta el host FTP? Utilice el nslookup para verificar.

- **Archivos Corrompidos**

¿UHa utilizado usted el modo correcto de transferencia?

- **FTP Anónimo**

Browsers genéricos, como son por ejemplo el Mozilla, Firefox, pueden que necesiten que la dirección de correo electrónico sea configurada. ¿Ha seguido usted todos los pasos para configurar su servidor?

DNS

Errores de DNS son comunes en muchas redes. En esta sección, enumeramos algunos de los problemas relacionados con los DNS y sus posibles remedios para corregirlos.

¿Ha configurado usted correctamente su resolución de nombres? Si es así, usted debe poder entrar en contacto con su servidor de nombres. Intente hacerle ping a su servidor de nombres para comprobarlo.

Ponga a prueba es espacio de nombre de dominio usando los siguientes comandos y utilidades:

- **nslookup**

```
$  nslookup
> Server
Default  Server:    codigolibre.org
Address:                 66.17.131.10
> set q=any
> redhat.com
Server:         10.0.0.1
Address:        10.0.0.1#53

Non-authoritative answer:
redhat.com      nameserver = ns2.redhat.com.
redhat.com      nameserver = ns3.redhat.com.
redhat.com      nameserver = ns1.redhat.com.

Authoritative answer can be found from:
```

- **Domain Internet Gropet (DIG)**

El comando dig (Domain informatioI Groper) constituye una herramienta para realizar consultas de diverso tipo a un servidor de DNS. Este muestra las respuestas recibidas de acuerdo a su solicitud. Es muy útil para detectar problemas en la configuración de los servidores de DNS debido a su flexibilidad, facilidad de uso y claridad en su salida.

Aunque normalmente las consultas que permite dig se definen en la línea de comando también se puede hacer en un fichero y pasárselo como argumento (opción -f). En el caso de que no se indique el servidor a consultar se asumirán los especificados en /etc/resolv.conf. Cuando no se añade ninguna opción o argumento en la línea de comando se consultan los servidores de nombres del dominio raíz (NS query).

La forma básica de invocar a dig es:

```
dig <@servidor> <nombre> [tipo]
```

donde:

- @servidor - es el nombre o la dirección IP del servidor a consultar.
- nombre - es el nombre de dominio del record por el cual se quiere preguntar.
- tipo - es el tipo del record por el que se consulta (ANY, NS, SOA, MX, etc.). De no indicarse se asumirá A.

Sintaxis: dig [@servidor] [opciones] [nombre] [tipo] [clase] [opciones de consulta]

Algunas opciones:

- -h : muestra la ayuda del comando.
- -x : hace consultas inversas, o sea, a partir de las direcciones IP determina nombres de dominio.
- -f <filename> : toma las consultas a partir de un fichero. Estas se definen una por línea y con la misma sintaxis que en la línea de comand

Ejemplos:

```
$ dig @alma maildi +search
; <<>> DiG 9.1.1 <<>> @alma maildi +search
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 6750
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1

;; QUESTION SECTION:
;maildi.disaic.cu.              IN      A
;; ANSWER SECTION:
maildi.disaic.cu.       86400   IN      A       192.168.100.3
;; AUTHORITY SECTION:
disaic.cu.              86400   IN      NS      alma.disaic.cu.
;; ADDITIONAL SECTION:
alma.disaic.cu.         86400   IN      A       192.168.100.2

;; Query time: 75 msec
;; SERVER: 192.168.100.2#53(alma)
;; WHEN: Sun Apr 29 17:25:05 2001
;; MSG SIZE  rcvd: 85

$ dig @192.168.200.3 shine.disaic.cu +short
; <<>> DiG 9.1.1 <<>> @192.168.200.3 shine.disaic.cu +short
;; global options:  printcmd
192.168.200.2

$ dig disaic.cu NS
...
;; ANSWER SECTION:
disaic.cu.              86400   IN      NS      alma.disaic.cu.
disaic.cu.              86400   IN      NS      odin.disaic.cu.
...

$ dig -x 192.168.200.4
...
;; QUESTION SECTION:
;2.100.168.192.in-addr.arpa.    IN      PTR
;; ANSWER SECTION:
4.200.168.192.in-addr.arpa. 86400 IN    PTR     sion.disaic.cu.
...
```

Compruebe sus archivos maestros. Puede ser que le falten algunos puntos al final de algunos FQDNs. Volque los datos, que puede ser que revisando encuentre un nombre de dominio u opción mal deletreada.

Errores del SMTP

Observe siempre la cabecera de retorno para ver donde el mensaje falló y adonde fué. Revise los host To: y Cc: existen con los comandos nslookup y con ping. El correo puede salir vía otro host, así que compruebe los records MX en el espacio de nombre de dominio. Compruebe que exista el usuario.

Intente efectuar un finger al host de destino. Intente un telnet al puerto 25 y use el comando VRFY, entonces use el comando EXPN para las listas de correo.

Intente enviar correo de prueba manualmente. Primero, telnet al puerto 25. Luego escriba los siguientes comandos “HELO”, “MAIL”, “RCPT”, y “DATA”. Si todo esto falla, envíe un correo al postmaster.

RESOLUCION DE PROBLEMAS TCP/IP

Las NICs (Tarjetas de Interfaz de Red) le ofrecen a los sistemas hablar uno con otro en una red. El nivel del hardware es el primero de la resolución de problemas en un NIC. Consulte el manual del usuario que viene con la tarjeta de red o el módem para asegurarse que esten correctamente instalado y configurados. Después de probar el hardware, un número de programas están disponibles para localizar averías enle interfaz incluyendo Ismod, insmod, ifconfig, ping, y tcpdump.

Los asuntos siguientes se discuten en esta sección:
- Módulos de Kernel
- Comandos de Resolución de Problemas de la Red
- Archivos Útiles de la Red
- ICMP (Internet Control Message Protocol)
- Otras Fuentes de Información

Módulos de y el Kernel

Antes de que usted pueda utilizar su interfaz de la red, el módulo correcto debe ser cargado en el núcleo para que reconozca la tarjeta. En GNU/Linux, los drivers están en forma de módulos del kernel. Para una vez que se cargue un módulo, hay un número de comandos para visualizar y manipular los módulos.

Hay muchos utilitarios para visualizar la información de configuración de la red y monitorear la infformación del hardware, pero el kernel proporciona una manera fácil de tener acceso a esta información atraves del sistema de archivos virtual proc en /proc.

En esta sección se discuten ambos los módulos del kernel y el sistema de archivos virtual proc.

Módulos del Kernel

Los comandos modprobe, lsmod, insmod y rmod son útiles para manipular módulos para instalar y configurar las interfaces de red. Todos estos comandos requieren la autenticación del super usuario. Escriba el siguiente comando para visualizar la información acerca de los módulos cargados:

#/sbin/lsmod

Este comando despliega la información del nombre, el tamaño, la cuenta del uso, y la lista de los módulos referidos.

Module	Size	Used by Not tainted	
appletalk	23884	12	(autoclean)

8139too	***17024***	***1***	
mii	***3976***	***0***	***(8139too)***
e100	***51428***	***1***	
ipt_LOG	***4416***	***1***	***(autoclean)***

El comando modprobe puede cargar un conjunto de módulos, un solo módulo, una pila de módulos dependientes o todos los módulos marcados con una etiqueta particular. El modprobe primero busca en los directorios que contienen módulos compilados para la versión más reciente del kernel de GNU/Linux.

El comando modprobe puede explorar para los módulos disponibles y carga automáticamente cualquier módulo bajo necesitado en un apilado del módulo. La línea de comando siguiente inicia el modprobe para enumerar todos los módulos disponibles:

/sbin/modprobe -l

El insmod instala un módulo cargable en el kernel en ejecusión. Este hace el intento de vincular (linkear) un módulo al núcleo corriente mirando la tabla de símbolo exportada del núcleo. El siguiente comando intenta instalar el módulo para la tarjeta de Ethernet Realtek 8139too:
/sbin/modprobe 8139too

El comando rmmod descarga los módulos cargado en el kernel. Este comando quita todos los módulos cargado en el kernel.
/sbin/modprobe -a

Después de instalar correctamente el módulo apropiado para el interfaz de la red, usted puede utilizar el comando ifconfig de configurar y de modificar los interfaces en tiempo real.

El Sistema de Archivos Virtual proc

El directorio /proc en un sistema GNU/Linux contiene los archivos virtuales creados por el kerne! de modo que usted y el sistema puedan supervisar qué está sucediendo actualmente en su sistema. Algunos de estos archivos son útiles para analizar y diagnosticar averías de interfaces de red.

Escriba la siguiente línea:
cat /proc/net/dev

Esto da salida a las líneas con la tarjeta Ethernet listada. Si la salida no es lo que se esperaba, entonces su kernel de GNU/Linux no sabe de la existencia de su adaptador de red. Una razón posible para que el kernel no reconozca la tarjeta Ethernet es porque tenga un conflicto con la interrupción o de dirección. Para ver una lista de las interrupciones y de las direcciones de I/O (Entrada/Salida), escriba los siguientes comandos:

cat /proc/interrupts
cat /proc/ioports

El comando cat /proc/interrupts producirá una salida similar a la siguiente:

```
[root@root]# cat /proc/interrupts
          CPU0
  0:  10269073        XT-PIC  timer
  1:         8        XT-PIC  keyboard
  2:         0        XT-PIC  cascade
  8:         1        XT-PIC  rtc
 11:   3876069        XT-PI
```

La interrupción es una señal enviada desde un dispositivo a la entrada del procesador de que un servicio es necesario. A menudo las interrupciones son llamados IRQs. La lista muestra que el adaptador de red eth0 situado en el interruptor 11 y que ningún otro dispositivo está compartiendo este número de la interrupción.

```
[root@proxy-ap root]# cat /proc/ioports
0000-001f    :  dma1
0020-003f    :  pic1
0040-005f    :  timer
0060-006f    :  keyboard
0070-007f    :  rtc
0080-008f    :  dma page reg
00a0-00bf    :  pic2
00c0-00df    :  dma2
00f0-00ff    :   fpu
0170-0177    : ide1
01f0-01f7    : ide0
02f8-02ff    : serial(auto)
0376-0376    : ide1
03c0-03df    : vga+
03f6-03f6    : ide0
03f8-03ff    : serial(auto)
0cf8-0cff    : PCI conf1
1000-101f    : Intel Corp. 82557/8/9 [Ethernet Pro 100]
  1000-101f  : e100
1020-103f    : Intel Corp. 82371AB/EB/MB PIIX4 USB
  1020-103f  : usb-uhci
1040-104f    : Intel Corp. 82371AB/EB/MB PIIX4 IDE
  1040-1047  : ide0
  1048-104f  : ide1
1400-14ff    : Accton Technology Corporation SMC2
```

La dirección 1000-101f de I/O esta asignada solamente al eth0, así demostrando que no hay ningún conflicto.

Comandos de Resolución de Problemas de Redes

Utilice los siguientes comandos para la resolución de problemas en la red:

- ifconfig
- ping
- tcpdump
- traceroute
- netstat
- netperf
- arp
- nslookup
- hostname

ifconfig

El comando Ifconfig es un programa que gestiona los dispositivos de red conectados a tu equipo y los enlaza con interfaces de red. Ejemplos de dispositivos de red posibles son tarjetas de red (interfaz ethX), modems (pppX), puerto paralelo (plipX), infrarrojos ... Una vez que tengamos la tarjeta de red bien instalada, (es decir, cargados los módulos correspondientes o compilado en el kernel el soporte para la tarjeta) tendremos que crear un interfaz que será por el que se envíe la información a los equipos conectados a

este dispositivo de red. Para esto ejecutamos el comando ifconfig.

El Ifconfig es una utilidad de línea de comandos que permite obtener y configurar las interfaces de red de un equipo. Si no se proporcionan argumentos, ifconfig muestra el estado de las interfaces de red que se encuentran activas. Si se proporciona una interfaz como argumento, ifconfig muestra el estado de dicha interfaz. Si se utiliza con la opción -a, muestra el estado de todas las interfaces, incluso aquellas que se encuentren desactivadas. Para configurar una interfaz se debe utilizar el formato:

ifconfig <interfaz> <familia> <dir_ip> netmask <máscara> broadcast <dir_broadcast> up

- **En interfaz**

  Debe proporcionarse el nombre de la interfaz de red que se desea configurar. Generalmente el nombre de interfaz se forma a partir de un nombre de manejador seguido de un número de unidad. El nombre de manejador para redes Ethernet es eth y las unidades comienzan a numerarse a partir de 0 hasta el número de interfaces existentes del mismo tipo menos uno (eth0, eth1, etc.).
- **En familia**

  Se debe proporcionar el nombre de una familia de direcciones soportada por el sistema. Este nombre se utilizará para decodificar y mostrar en un formato inteligible todas las direcciones de protocolo. Las familias de direcciones más comúnmente utilizadas son inet para TCP/IP, inet6 para IP versión 6 e ipx para Novell IPX.
- **dir_ip**

  Es la dirección IP con que se desea configurar la interfaz de red.
- **máscara**

  Establece la máscara de red que se desea utilizar para la interfaz. Si no se proporciona este valor, se utilizarán las máscaras de red por defecto para direcciones clase A, B o C en función de la dirección IP con que se esté configurando esta interfaz.
- **dir_broadcast**

  Al proporcionar una dirección de broadcast con la opción broadcast, se indica a la interfaz de red que se desea que habilite el modo de broadcast dirigido y que contemple dicha dirección. La dirección de broadcast dirigido a una red se determina a partir de la dirección IP de cualquiera de los equipos pertenecientes a dicha red y su máscara de red, ya que se forma a partir la dirección de red poniendo '1's en la parte de dirección correspondiente a equipos.

El comando ifconfig presenta muchas otras opciones que pueden ser consultadas mediante la ayuda en línea de GNU/Linux (***man ifconfig***). Una de ellas, útil para el desarrollo de la práctica, es la opción mtu valor, que permite establecer la unidad máxima de transferencia (MTU, del inglés Maximum Transfer Unit) que se desea utilice la interfaz que está configurándose. Antes de reconfigurar una interfaz de red, es conveniente desactivarla utilizando la opción down. Utilice para ello el formato ***ifconfig interfaz down***

Para ver un listado de los diversos interfaces se demuestra aquí debajo:

```
k1k1@kikla:~$ /sbin/ifconfig
eth0    Link encap:Ethernet  HWaddr 00:0B:CD:36:04:2A
        inet6 addr: fe80::20b:cdff:fe36:42a/64 Scope:Link
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:77 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 b)  TX bytes:23670 (23.1 KiB)
        Interrupt:10 Base address:0x4000

eth1    Link encap:UNSPEC  HWaddr 00-0B-CD-71-A0-35-D7-22-00-00-00-00-00-00-00-00
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:71 dropped:71 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
```

```
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

eth2      Link encap:Ethernet  HWaddr 00:09:5B:54:89:26
          inet addr:10.0.0.106  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::209:5bff:fe54:8926/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7153 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5212 errors:5 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3487577 (3.3 MiB)  TX bytes:1293400 (1.2 MiB)
          Interrupt:3 Base address:0x100

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:167 errors:0 dropped:0 overruns:0 frame:0
          TX packets:167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:28096 (27.4 KiB)  TX bytes:28096 (27.4 KiB)

sit0      Link encap:IPv6-in-IPv4
          inet6 addr: ::127.0.0.1/96 Scope:Unknown
          UP RUNNING NOARP  MTU:1480  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:69 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Si el PPP esta configurado en el sistema, será incluido en la lista de interfaces. Los siguientes son algunos comandos que usted puede utilizar para ver si la dirección IP y mascaras de red son las correcta asociadas con la tarjeta de Ethernet.

El formato del comando es:
/sbin/ifconfig interface opciones

Una de las aplicaciones del comando ifconfig es el de iniciar y detener una interface Ethernet usando la opciones up y down.

Por ejemplo:
/sbin/ifconfig eth0 up
/sbin/ifconfig eth0 down

Si falta el interfaz del loopback, escriba:
/sbin/ifconfig lo 127.0.0.1

Si le falta el adaptador de la red (tarjeta de Ethernet), escriba:
ifconfig eth0 192.168.2.7 up
La dirección IP interna ahora es asociada con esta tarjeta de Ethernet.

Para incluir un mascara de red, escriba:
ifconfig eth0 192.168.2.7 netmask 255.255.255.0

Si la dirección de red es una dirección de la clase C, el ifconfig asignará automáticamente la netmask 255,255,255.0.

Si estos comandos todavía no consiguen ingresar la computadora en la red, entonces investigue dentro del kernel de su GNU/Linux para ver qué dispositivos tiene.

El comando ping

El utilitario ping usa datagramas de ECHO_REQUEST para obtener un ICMP ECHO_RESPONSE de un host o de una pasarela (gateway). Los ping incluyen el número de bytes devueltos desde la interfaz, la secuencia y el tiempo de ida y vuelta. Esta utilidad proporciona el mejor método para confirmar si un adaptador de red está configurado correctamente y si el enrutamiento esta establecido correctamente. La primera cosa quedebemos hacer es ping al interfaz de loopback.

Escriba el siguiente comando para efectuar un ping al interfaz loopback:

ping 127.0.0.1

Esto debe producir una respuesta similar a la siguiente:

```
k1k1@kikla:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.067 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.079 ms
--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.067/0.073/0.079/0.005 ms
```

El ping puede también seguir tráfico de la red en otras máquinas conectadas a la red. El comando ping imprime una línea por cada respuesta recibida. El comando ping continuará recibiendo paquetes hasta que el usuario termine (generalmente por un CONTROL+C). El comando ping es útil para observar el tiempo de respuesta de los paquetes o ver si se saltan las secuencias de los números en serie. Estos datos son importantes en la investigación de pérdida de paquete y tráfico de la red.

En la siguiente tabla se enumeran varias opciones del comando ping.

Opción	**Descripción**
-c #número	Detiene el ping después de enviar y recibir #número de paquetes ping.
-f flood ping	Da salida de paquetes ping lo más rápido posible o 100 por segundo. Sólo el root puede usar esta opción.
-i wait	Esta opción espera (waits) los segundos indicados por el valor wait entre los paquetes enviados.
-s patketsize	Esta específica el número de bytes a ser enviado. Por defecto es de 56, pero con el cabezal ICMP la data se con vierte en 64 bytes.

El Comando tcpdump

El Tcpdump es un programa cuya utilidad principal es analizar el tráfico que circula por la red. Se apoya en la librería de captura pcap, la cual presenta una interfaz uniforme y que esconde las peculiaridades de cada sistema operativo a la hora de capturar tramas de red. Para seguir el manual es necesario unos conocimientos basicos del protocolo TCP/IP, remitiéndome al TCP/IP Illustrated, Volumen 1 de Stevens, para quien esté interesado.

Aunque viene incluido con la mayoría de las distribuciones de Linux, sus fuentes pueden encontrarse en www.tcpdump.org. Lo ejecutamos simplemente escribiendo:

tcpdump

Pero esto no es lo normal, empecemos a introducir opciones y comentar para que es cada una. Algunos ejemplos rapidos de tcpdump:

tcpdump –i eth0 –c 100 –s 500

Nos sacará por pantalla los primeros 100 paquetes (-c 100) que pasen por la interfaz eth0 (-i eth0) con un tamaño maximo de paquete de 500 bytes (-s 500)

tcpdump –qec 1

Tenemos varias formas de sacar paquetes por pantalla: -q (quiet o silencioso (poca información)) y –v y –vv que van de menos a más información. La opción –e nos sacará las direcciones mac origen/destino y –c 1 nos sacará solo el primer paquete como ya hemos explicado antes.

Que tipo de tráfico podemos capturar: Podemos capturar tráfico basado en:

- Direcciones
- Protocolos
- Puertos
- Características de paquetes
- Combinacion de todos estos

Filtrando por Direcciones

tcpdump host ivelis

Nos sacará toda la información relativa al host '***ivelis***'.

tcpdump dst host ivelis

Nos sacará toda la información donde el host destino sea 'ivelis'

tcpdump ether src host 0:a0:3b:3:e1:1d

Nos sacará toda la información donde la tarjeta de red de origen sea la dirección mac: 0:a0:3b:3:e1:1d (Sintaxis: ether host)

tcpdump -c 100 net 192.168.1.0 mask 255.255.255.0

Mostrará los primeros 100 paquetes (-c 100) de la red 192.168.1.0 (net 192.168.1.0) con una mascara de red de 255.255.255.0 (mask 255.255.255.0)

Filtrando por Protocolos

tcpdump upd

Mostrará todos los paquetes que viajen por udp

tcpdump –i eth0 –c 1000 –s 300 udp

Mostrará los primeros 1000 (-c 1000) paquetes de la interfaz eth0 (-i eth0) con un tamaño de paquete maximo de 300 bytes (-s 300) en el protocolo udp.

tcpdump port 23

Mostrará todos los paquetes que vayan por el puerto 23

Mezclando filtros

tcpdump not port 22

Mostrará toda la información excepto la que vaya por el puerto 22

tcpdump port 23 and host ivelis

Mostrará toda la información que vaya por el puerto 23 y por el host ivelis

tcpdump not "(port 23 and host ivelis and host miguel)"

Mostrará toda la información que ni vaya por el puerto 23, ni por el host ivelis ni por el host miguel

tcpdump dst host ivelis and dst port 80

Mostrará toda la información que vaya al host ivelis y al puerto destino 80 (dst port 80)

tcpdump –i etho –c 100 –s 1000 src host miguel and "(dst host ivelis or dst host crithian)" and dst port 22

Como puede apreciar, mezclando información podemos tener filtros largos y todo lo complicados que queramos… este filtro nos mostrará los primeros 100 paquetes que pasen por la interfaz eth0 (-i eth0) con tamaño maximo de paquete de 1000 bytes (-s 1000) que vengan del host miguel (src host miguel) y que vayan o al host ivelis o al host cristhian "(dst host ivelis or dst host cristhian)" al puerto 22 (dst port 22).

El Comando traceroute

Esta orden se utiliza para imprimir la ruta que los paquetes siguen desde nuestro sistema hasta otra máquina; para ello utiliza el campo TTL (Time To Live) del protocolo IP, inicializándolo con valores bajos y aumentándolo conforme va recibiendo tramas ICMP de tipo TIME/SMALL>_EXCEEDED. La idea es sencilla: cada vez que un paquete pasa por un router o una pasarela, esta se encarga de decrementar el campo TTL en una unidad; en el caso de que se alcance un valor 0, se devuelve un paquete TIME/SMALL>_EXCEEDED y se descarta la trama. Así, traceroute inicializa a 1 este campo, lo que ocasiona que el primer router encontrado ya devuelva el mensaje de error; al recibirlo, lo inicializa a 2, y ahora es el segundo router el que descarta el paquete y envía otro mensaje de error, y así sucesivamente. De esta forma se va construyendo la ruta hasta un determinado host remoto:

```
# traceroute www.codigolibre.org
 1  * * *
 2  209.58.3.52 (209.58.3.52)  20.45 ms  25.05 ms  31.593 ms
 3  192.168.102.1 (192.168.102.1)  17.098 ms  15.996 ms  15.178 ms
 4  172.22.0.205 (172.22.0.205)  15.444 ms  17.175 ms  15.143 ms
 5  172.22.0.253 (172.22.0.253)  24.353 ms  25.77 ms  24.13 ms
 6  172.22.0.133 (172.22.0.133)  21.846 ms  26.21 ms  21.839 ms
 7  hme0-bavaro.codetel.net.do (196.3.74.5)  49.477 ms  49.497 ms  48.532 ms
 8  ge-1-0-0.r01.miamfl02.us.bb.verio.net (129.250.11.101)  52.83 ms  50.249 ms  50.725 ms
 9  p4-0-0-0.r00.miamfl02.us.bb.verio.net (129.250.3.184)  48.745 ms  50.081 ms  48.285 ms
10  p4-4-2-0.r01.dllstx09.us.bb.verio.net (129.250.5.58)  92.83 ms  96.722 ms  94.801 ms
11  so-2-3-0.edge2.dallas1.level3.net (4.68.127.41)  93.029 ms  100.019 ms  92.208 ms
12  so-7-0-0.bbr2.dallas1.level3.net (4.68.96.121)  110.301 ms  117.371 ms  112.163 ms
13  4.68.122.11 (4.68.122.11)  88.584 ms 4.68.122.43 (4.68.122.43)  86.717 ms  87.021 ms
14  4.78.224.74 (4.78.224.74)  87.983 ms  90.102 ms  86.997 ms
15  206.123.64.42 (206.123.64.42)  90.201 ms  88.392 ms  90.278 ms
16  abiertos.org (206.123.67.148)  86.21 ms  87.712 ms  86.113 ms
```

El comando traceroute se utiliza para realizar pruebas, medidas y administración de una red; introduce mucha sobrecarga, lo que evidentemente puede acarrear problemas de rendimiento, llegando incluso a negaciones de servicio por el elevado tiempo de respuesta que el resto de aplicaciones de red pueden presentar. Además, se trata de un programa contenido en un fichero setuidado, por lo que es interesante resetear el bit de setuid de forma que sólo el root pueda ejecutar la orden: hemos de pensar que un usuario normal rara vez tiene que realizar pruebas sobre la red, por lo que el bit setuid de traceroute no es más que un posible problema para nuestra seguridad; aunque con ping sucede lo mismo (es un fichero setuidado), que un usuario nece-

site ejecutar traceroute es menos habitual que que necesite ejecutar ping (de cualquier forma, también podríamos resetear el bit setuid de ping).

El formato de comando para Linux es:
$ traceroute dirección_ip
En el ejemplo, dirección_ip identifica el sistema remoto. En la sguiente tabla enumeramos varias opciones de la línea de comandos.

Opción	**Descripción**
-f	Establece el valor TTL inicial en el primer paquete saliente
-F	Estable el bit del valor "don't fragment"
-g host-list	Específica la ruta a un gateway "loose-source"
-l	Opción usa ICMP ECHO en ves de datagramas UDP
-m	Específica el máximo número de saltos que el utilitario buscará; por defecto es 30
-n	Nombre del host no se resolverá a direcciones en cada salto .
-p	Establece el número de puerto base UDP usado en las pruebas (por defecto es 33434).
-q	Número de consultas para el número de peticiones del número de paquetes que deben ser enviados por cada valor TTL para recibir el valor RTT por cada salto
-r	Sobre pasa las tablas de enrutamiento normal y envía directamente a un host conecta do a la red.
-s	Utiliza la dirección IP proveída en los paquetes de prueba que envía
-t	Específica el valor ToS en los paquetes de prueba a enviar
-w timeout	Específica el tiempo en segundos a esperar por cada respuesta. Por defecto es 5 segundos.

Ejemplos de traceroute desde la línea de comando:
traceroute nombre-host
dará lugar a la siguiente respuesta de un solo salto debido a que está en la misma subnet:
$ traceroute ivelis
traceroute to ivelis (10.0.0.11), 30 hops max, 40 byte packets
1 ivelise (10.0.0.11) 22.323 ms 2.63 ms 0.707 ms

Si no hay respuesta de un enrutador dentro de un intervalo timeout de 5 segundos, se imprime una "*" para ese paquete de prueba. Este intervalo de timeout se puede cambiar con la opción -w según se explica en la tabla de opciones.

El comando traceroute es para el uso de encontrar fallas en la red, y debe ser usado sólo por la gerencia. Debe ser utilizado sobre todo para el aislamiento manual de fallas ya que este solamnete procura seguir la trayectoria que viaja el paquete. Este impone una carga adicional a la red con los paquetes a la destinación con diversos valores TTL y por lo tanto, no es sabio utilizar el traceroute durante tiempo normales de operaciones o escribir scripts de tareas automatizadas ya que inundan la red. Impone una carga adicional ante la red con los paquetes a la destinación con diversos valores de la TTL, y por lo tanto, no es sabio utilizar el traceroute durante operaciones normales o de las escrituras automatizadas pues inunda la red.

El Comando netstat

Esta orden se utiliza para visualizar el estado de diversas estructuras de datos del sistema de red, desde las tablas de rutado hasta el estado de todas las conexiones a y desde nuestra máquina, pasando por las tablas ARP, en función de los parámetros que reciba.

En temas referentes a la seguridad, netstat se suele utilizar, aparte de para mostrar las tablas de rutado de ciertos sistemas (con la opción -r, como hemos visto antes), para mostrar los puertos abiertos que escuchan peticiones de red y para visualizar conexiones a nuestro equipo (o desde él) que puedan salirse de lo habitual. Veamos un ejemplo de información mostrada por netstat:

netstat -P tcp -f inet -a
TCP

Local Address	Remote Address	Swind	Send-Q	Rwind	Recv-Q	State
.	*.*	0	0	0	0	IDLE
*.sunrpc	*.*	0	0	0	0	LISTEN
.	*.*	0	0	0	0	IDLE
*.32771	*.*	0	0	0	0	LISTEN
*.ftp	*.*	0	0	0	0	LISTEN
*.telnet	*.*	0	0	0	0	LISTEN
*.finger	*.*	0	0	0	0	LISTEN
*.dtspc	*.*	0	0	0	0	LISTEN
*.lockd	*.*	0	0	0	0	LISTEN
*.smtp	*.*	0	0	0	0	LISTEN
*.8888	*.*	0	0	0	0	LISTEN
*.32772	*.*	0	0	0	0	LISTEN
*.32773	*.*	0	0	0	0	LISTEN
*.printer	*.*	0	0	0	0	LISTEN
*.listen	*.*	0	0	0	0	LISTEN
*.32774	*.*	0	0	0	0	LISTEN
.	*.*	0	0	0	0	IDLE
*.6000	*.*	0	0	0	0	LISTEN
*.32775	*.*	0	0	0	0	LISTEN
localhost.32777	localhost.32775	32768	0	32768	0	ESTABLISHED
localhost.32775	localhost.32777	32768	0	32768	0	ESTABLISHED
localhost.32780	localhost.32779	32768	0	32768	0	ESTABLISHED
localhost.32779	localhost.32780	32768	0	32768	0	ESTABLISHED
localhost.32783	localhost.32775	32768	0	32768	0	ESTABLISHED
localhost.32775	localhost.32783	32768	0	32768	0	ESTABLISHED
localhost.32786	localhost.32785	32768	0	32768	0	ESTABLISHED
localhost.32785	localhost.32786	32768	0	32768	0	ESTABLISHED
localhost.32789	localhost.32775	32768	0	32768	0	ESTABLISHED
localhost.32775	localhost.32789	32768	0	32768	0	ESTABLISHED
localhost.32792	localhost.32791	32768	0	32768	0	ESTABLISHED
localhost.32791	localhost.32792	32768	0	32768	0	ESTABLISHED
localhost.32810	localhost.6000	32768	0	32768	0	ESTABLISHED
localhost.6000	localhost.32810	32768	0	32768	0	ESTABLISHED
anita.telnet	luisa.2039	16060	0	10136	0	ESTABLISHED
anita.telnet	ltorvalds.linux.org.1068	15928	0	10136	0	ESTABLISHED
localhost.32879	localhost.32775	32768	0	32768	0	TIME_WAIT
.	*.*	0	0	0	0	IDLE

Por un lado, en este caso vemos que hay bastantes puertos abiertos, esto es, escuchando peticiones: todos los que presentan un estado LISTEN, como telnet, finger o smtp (si es un servicio con nombre en /etc/services se imprimirá este nombre, y si no simplemente el número de puerto). Cualquiera puede conectar a este servicio (como veremos en el siguiente punto) y, si no lo evitamos mediante TCP Wrappers, utilizarlo para enviarle peticiones.

Aparte de estos puertos a la espera de conexiones, vemos otro gran número de conexiones establecida entre nuestro sistema y otros (como antes hemos dicho, desde nuestro equipo o hacia él); casi todas las esta-

blecidas (estado ESTABLISHED) son de nuestra máquina contra ella misma, lo que a priori no implica consecuencias de seguridad. Otra de ellas es desde un equipo de la red local contra nuestro sistema, lo que también es bastante normal y no debe hacernos sospechar nada; sin embargo, hay una conexión que sí puede indicar que alguien ha accedido a nuestro sistema de forma no autorizada: si nos fijamos, alguien conecta por telnet desde la máquina ltorvalds.linux.org. Es raro que tengamos a un usuario allí, por lo que deberíamos monitorizar esta conexión y las actividades que esta persona realice; es muy probable que se trate de alguien que ha aprovechado la inseguridad de ciertos sistemas para utilizarlos como plataforma de ataque contra nuestros UNiX.

La tabla siguiente enumera las opciones comúnmente usadas con el comando del netstat.

Opción	Descripción
-a	Muestra el estado de todos los sockets
-c	Refresca toda la información cada segundo
-n	Reporta las direcciones de Internet y los números de puertos como números y no como nombre de hosts ni como servicios
-r	Despliega las tablas de enrutamiento del sistema
-s	Provée estadisticas de los paquetes procesados por su sistema
-i	Muestra el estado de las interfaces que han sido autoconfiguradas o una interface particular si es especificada
-1	Despliega los puertos actualmente escuchando y los sockets

El comando del netstat de Linux proporcionará mucha estadística sin ningunas opciones. Por ejemplo, desde la línea de comandos de GNU/Linux escriba:

netstat

Esto le dará una respuesta (sólo se presenta una parta para breviedad) como la siguiente:

```
k1k1@kikla:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q   Send-Q   Local Address            Foreign Address            State
tcp     0        0       10.0.0.182:43302        abiertos.org:www           ESTABLISHED
tcp     0        0       10.0.0.182:43298        abiertos.org:www           ESTABLISHED
udp     0        0      localhost.localdom:1024  localhost.localdom:1024    ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type     State       I-Node   Path
unix  10     [ ]     DGRAM                5645     /dev/log
unix  2      [ ]     DGRAM                5852     @/var/run/hal/hotplug_socket
unix  2      [ ]     DGRAM                1746     @udevd
unix  2      [ ]     DGRAM                12248
unix  3      [ ]     STREAM   CONNECTED   10382    /tmp/.ICE-unix/4605
unix  3      [ ]     STREAM   CONNECTED   10381
unix  3      [ ]     STREAM   CONNECTED   10375    /tmp/.X11-unix/X0
```

El Comando netperf

Netperf es un benchmark (prueba de rendimiento) que puede ser usada para localizar averías y problemas en una red. La información proporcionada por netperf puede ser utilizada para localizar congestión en una red y entonces se pueden tomar las medidas apropiadas para aliviar el problema.

El programa netperf proporciona información en una gran variedad de asuntos, incluyendo ambas prueba de flujo y de petición/respuesta y puede usar varios diversos protocolos de red, incluyendo el TCP y UDP. Un factor importante que se debe observar es que para probar una interfaz de red con excepción de la interfaz loopback, lo, el netperf debe estar instalado en otra máquina en la red.

Hay dos programas separados que conforman a netperf, el programa cliente, nerperf, que es el responsa-

ble de iniciar una conexión a la otra parte que es el servidor de netperf, llamado netserver. Puesto que netperf contiene el netserver, hay dos maneras posibles de instalarlo. Para los usuarios con el acceso a la cuenta de root, el netserver se puede instalar como demonio invocado por el demonio de Internet, el inetd. Si no, el netserver se puede instalar como demonio independiente.

Usar netperf

Como se mencionó previamente, netperf puede realizar diversos tipos de pruebas. Los tipos más comunes de pruebas lo más probable serán para TCP y UDP. Hay varias opciones de la línea de comandos para cambiar los tipos y las especificaciones de las pruebas que se desean realizar. Algunas de las opciones de la línea de comandos son:

-s size	Específica el tamaño del socket del buffer local de enviar y recibir
-S size	Efectuá la operación de -s para la máquina remota
-m size	Envia los paquetes de tamaño size en bytes
-M size	Recibe paquetes de tamaño size en bytes; como -m para sistema remoto
-t test	Específica el tipo de prueba a efectuar
-H host	Se conecta a host para poner a prueba la red
-l length	Efectuá la prueba por length secongundos
-h	Desplega un mensaje de uso

El comando /opt/netperf/netperf produce una prueba del flujo TCP. Esta prueba por general se realiza en el interfaz del loopback y tiene una duración de 10 segundos. La prueba envía un mensaje de un puerto a otro. El tamaño del mensaje se mide en el puerto de inicio en el puerto final. El rendimiento de procesamiento se mide en kilobytes por segundo. Aquí le presentamos un método rápido para comprobar si el programa del netperf está trabajando correctamente:

k1k1@kikla:~$ /opt/netperf/netperf -H host -t type

Aquí el host es el destino y los tipos de pruebas son especificados por host y type (tipo), respectivamente. Para poder conectarse con el otro host, es importante recordar que el host de destino debe estar ejecutando netserver. Algunas de las pruebas que se pueden especificar con la opción -t son:

UDP_STREAM	Efectuá una prueba de Flujo UDP
TCP_STREAM	Efectuá una prueba de Flujo TCP
UDP_RR	Efectuá una prueba de Petición/Respuesta UDP
TCP_RR	Efectuá una prueba de Petición/Respuesta TCP

Hay varias otras pruebas que pueden ser realizadas por netperf; las que están enumerados anteriormente son generalmente las más comunes.

El Comando arp

El comando arp exhibe y modifica las tablas de conversión de dirección de Internet-a-Ethernet usadas por el ARP (Address Resolution Protocol). El formato de comando es: ***# arp options***

Cuando esté es utilizado sin ningunas opciones, el comando arp exhibe la información actualmente almacenada. El siguiente es un ejemplo de como usar el comando arp con su resultado:

```
kikla:/home/k1k1# arp
Address          HWtype  HWaddress           Flags Mask      Iface
10.0.0.1         ether   00:08:C7:F3:E6:E4   C               eth2
```

La opción -a de arp despliega la información en el formato raw:

kikla:/home/k1k1# arp -a
? (10.0.0.1) at 00:08:C7:F3:E6:E4 [ether] on eth2

Puede suprimir entradas del cache de arp usando la opción -d y especificando un nombre de host:

kikla:/home/k1k1# arp -d nombre_host

El Comando nslookup

El comando nslookup es un programa interactivo para efectuar preguntas a los servidores de nombre de dominio del Internet. El usuario tiene la opción de solicitar a un servidor de nombres en específico para que le proporcione la información sobre un host en particular o para conseguir una lista de todos los host en un dominio.

El formato del comando nslookup es:

nslookup opciones dirección

El siguiente es un ejemplo de como usar el comando nslookup:

```
kikla:/home/k1k1# nslookup miguel
Server:          usuarios.codigolibre.org
Address:         192.168.2.53
Name:            miguel.codigolibre.org
Address:         192.168.2.129
```

Para obtener una lista de todos los nodos en un dominio en particular, ejecute la siguiente secuencia de comandos:

```
# nslookup
> ls codigolibre.org
[ns.codigolibre.org]
$ORIGIN  codigolibre.org
@          ID     IN     A     192.168.2.254
usuarios   ID     IN     A     192.168.2.53
miguel     ID     IN     A     192.168.2.125
ns         ID     IN     A     192.168.2.254
>exit
```

El Comando hostname

El comando hostname imprime el nombre del host actual, parecido al presentado por el prompt del sistema, si así esta configurado, al inició de sesión o login. Aquí se le presenta un ejemplo de usar el comando hostname:

```
kikla:/home/k1k1# hostname
kikla
```

Ejercicio 5-1: Instale y Configure una tarjeta de Ethernet Como Interfaz de Red

En este ejercicio vamos a instalar correctamente una tarjeta Ethernet como interfaz de red. En este ejercicio, la tarjeta Ethernet es una 3COM 3C905. No se proporcionan soluciones para este ejercicio.

1. Ingrese al sistema como root.

2. Para buscar el módulo para utilizar para el dispositivo de red 3COM 3C905, escriba el siguiente comando para buscar el módulo apropiado:

   /sbin/modprobe -l

   Para enumerar los módulos una página a la vez, escriba:

/sbin/modprobe -l | more

3. El módulo en este caso será 3c59x.o. Escriba lo siguiente para instalar el módulo:
/sbin/insmod 3c59x

4. Para ver si el módulo ha sido instalado correctamente, digite:
/sbin/lsmod

5. Después de asegurarse que el módulo está presente, utilize el comando ifconfig para configurar la tarjeta de Ethernet. Asegúrese de que la tarjeta Ethernet está alias al interfaz correcto (eth0, eth1, eth2, etc.) en /etc/conf.modules en el archivo, sólo debe escribir:
cat /etc/conf.modules

Si el interfaz no es el deseado, utilice un editor de texto, como pico y modifiquélo. Ahora utilice el ifconfig para asignarle una Dirección IP:
/sbin/ifconfig eth0 192.168.2.15 up
Esta sentencia asigna la IP ADDRESS 192.168.2.7 a la interfaz eth0 y la activa (up).

6. Ahora es necesario manipular la tabla de enrutamiento para establecer las rutas estáticas en la red:
/sbin/route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0

Esta sentencia establece una ruta a la red 192.168.2.x atraves de la interfaz eth0

7. Por último utilice el comando ifconfig para ver si la tarjeta Ethernet ahora se está presentando activa en la red:
/sbin/ifconfig

Archivos de Red Utiles

Todo administrador de una red TCP/IP debe conocer bien los siguientes archivos de red. Entre los archivos claves de la red se incluyen:

• protocols
• services
• inetd.conf

El Archivo protocols

Cada línea del archivo protocolos describe los protocolos usados en la capa de Internet de la pila del protocolo TCP/IP según lo definido por RFC 1700 (Assigned Numbers). Cada línea se ajusta al siguiente formato:

nombre-oficial-del-protocolo protocol # aliases

El archivo de los protocolos es almacenado en el directorio /etc y se puede corregir fácilmente usando un editor de texto. Si usted desarrolla su propio protocolo que utiliza la interface raw de sockets, este deberá ser listado en el archivo protocol(s).

El número listado para cada protocolo se incluye en el campo Protocolo del cabezal IP. Ese número define el próximo nivel de protocolo para recibir el campo Data en el destino. El archivo exhibe la siguiente información (seleccionada), que variará dependiendo de la personalización para requisitos particulares:

$ cat /etc/protocols
#
Internet protocols

```
#
# $GNU/Linux:src/etc/protocols,v 1.14 2000/09/24 11:20:27 asmodai Exp $
#       from: @(#)protocols      5.1 (Berkeley) 4/17/89
#
# See also http://www.isi.edu/in-notes/iana/assignments/protocol-numbers
#
ip      0       IP              # Internet protocol, pseudo protocol number
#hopopt 0       HOPOPT          # hop-by-hop options for ipv6
icmp    1       ICMP            # Internet control message protocol
igmp    2       IGMP            # Internet group management protocol
ggp     3       GGP             # gateway-gateway protocol
```

El Archivo services

El archivo services contiene los números de los puertos de los servicios well-known (servicios bien conocidos) según están definidos por el RFC 1700 (Assigned Numbers). Cualquier aplicación que desea reservar un número de puerto debe especificarlo en el archivo services.

NOTA: Con simplemente listar una aplicación en el archivo de los servicios no significa que la aplicación estará soportada. El nombre del servicio y el número del puerto son una parte pequeña solamente de una aplicación funcional.

El formato de cada línea es como sigue:
nombre-del-servicio   puerto/protocolo   aliases # descripción/comentarios

El archivo de los servicios es almacenado en el direcrtorio /etc y se puede corregir fácilmente usando cualquier editor de texto.

Los servicios tales como telnet, ftp y TFTP son definidos en este archivo. Examinando este archivo, usted puede determinar en que números de puertos es que estos servicios están disponibles. El archivo exhibe la siguiente información (parcial), que variará dependiendo de sus requisitos particulares:

```
#
# Network services, Internet style
#
# WELL KNOWN PORT NUMBERS
#
rtmp            1/ddp           #Routing Table Maintenance Protocol
tcpmux          1/udp           # TCP Port Service Multiplexer
tcpmux          1/tcp           # TCP Port Service Multiplexer
qotd            17/udp          # Quote of the Day
qotd            17/tcp          # Quote of the Day
                                # Jon Postel <postel@isi.edu>
msp             18/udp          # Message Send Protocol
ftp-data        20/tcp          # File Transfer [Default Data]
ftp             21/udp          # File Transfer [Control]
ftp             21/tcp          # File Transfer [Control]
                                # Jon Postel <postel@isi.edu>
ssh             22/udp          # SSH Remote Login Protocol
ssh             22/tcp          # SSH Remote Login Protocol
telnet          23/tcp          # Telnet
```

```
smtp        25/udp      # Simple Mail Transfer
msg-auth    31/udp      # MSG Authentication
dsp         33/udp      # Display Support Protocol
time        37/udp      # Time
                        # Christopher Leong <leong@kolmod.mlo.dec.com>
finger      79/udp      # Finger
http        80/udp      www www-http # World Wide Web HTTP
kerberos    88/tcp      # Kerberos
sgmp        153/tcp     # SGMP
irc         194/tcp     # Internet Relay Chat Protocol
at-8        208/tcp     # AppleTalk Unused
ldap        389/tcp     # Lightweight Directory Access Protocol
tserver     450/tcp     # Computer Supported Telecomunication Applications
rcp         469/tcp     # Radio Control Protocol
pov-ray     494/tcp     # POV-Ray
login       513/tcp     # remote login a la telnet;
who         513/udp     # maintains data bases showing who’s
shell       514/tcp     # cmd
syslog      514/udp     #
printer     515/udp     # spooler
talk        517/udp     # like tenex link, but across
socks       1080/udp    # Socks
tripwire    1169/udp    # TRIPWIRE
tripwire    1169/tcp    # TRIPWIRE
nessus      1241/udp    # nessus
nessus      1241/tcp    # nessus
```

El Archivo inetd.conf

El archivo inetd.conf es en almacenado en el directorio /etc y se puede corregir fácilmente usando un editor de texto. El demonio inetd maneja a los demonios de los servicios de red TCP/IP según la información contenida en el archivo inetd.conf. Para reconfigurar los procesos ejecutándose en inetd, edite el archivo inetd.conf y envíele una señal de SIGHUP al inetd.

Cada línea en el archivo contiene la información requerida para manejar a un demonio de red en particular. El archivo exhibe la siguiente información (parcial), la cual variará dependiendo de la personalización para satisfacer los requisitos particulares:

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
#:INTERNAL: Internal services
#finger     stream   tcp   nowait   nobody   /usr/libexec/tcpd   fingerd -s
#ftp        stream   tcp   nowait   root     /usr/libexec/tcpd   ftpd -l
#login      stream   tcp   nowait   root     /usr/libexec/tcpd   rlogind
#nntp       stream   tcp   nowait   usenet   /usr/libexec/tcpd   nntpd
#ntalk      dgram    udp   wait     root     /usr/libexec/tcpd   ntalkd
#shell      stream   tcp   nowait   root     /usr/libexec/tcpd   rshd
#telnet     stream   tcp   nowait   root     /usr/libexec/tcpd   telnetd
#uucpd      stream   tcp   nowait   root     /usr/libexec/tcpd   uucpd
#comsat     dgram    udp   wait     root     /usr/libexec/tcpd   comsat
#tftp       dgram    udp   wait     nobody   /usr/libexec/tcpd   tftpd /private/tftpboot
```

```
#bootps    dgram    udp  wait    root    /usr/libexec/tcpd  bootpd
```

Control de Mensaje de Protocolos de Internet (ICMP)

El ICMP es el protocolo de la resolución de problemas de TCP/IP y es una parte requerida de la pila del TCP/IP. El ICMP se específica en RFC 792. Este permite que los host y las pasarelas/gateways del Internet reporten los errores a través de mensajes ICMP. Estos mensajes son encapsulados en la porción de datos del paquete IP y en última instancia son orientados al módulo de software del IP en el sistema de origen.

Cuando un datagrama viaja hacia su destino tendrá que atravesar una serie de pasarelas (routers), las cuales procesan el datagrama para dirigirlo adecuadamente hacia su destino. Si una pasarela no puede encaminar ese datagrama, o bien detecta alguna condición especial en la que se ve incapacitada para hacerlo (congestión de red, líneas fuera de servicio, etc.), entonces ese datagrama se pierde.

Estas y otras circunstancias en el tratamiento de los datagramas en su viaje hacia el destino, hacen necesario la creación de un mecanismo que, al menos, informe de estas situaciones al host origen, para que sea consciente de los problemas que ha sufrido el datagrama que ha enviado y, si procede, tome las acciones oportunas. De aquí nace el protocolo ICMP.

El protocolo ICMP (Internet Control Message Protocol), es un mecanismo que informa de la aparición de errores en la manipulación de los datagramas. Siempre que una pasarela detecte un error o excepción en un datagrama, utiliza el protocolo ICMP para informar al host origen de la circunstancia. ICMP no realiza ninguna acción para corregir el error que se haya producido, solamente se encarga de comunicarlo al host origen para que éste realice las acciones oportunas para corregir el error.

Originalmente, ICMP fué diseñado como un protocolo para los routers, sin embargo los host también lo pueden utilizar. Los mensajes ICMP van encapsulados en datagramas IP. El destino del mensaje ICMP no será la aplicación del usuario en el host destino, sino que deberá ser interpretarlo por su módulo ICMP. Si un mensaje ICMP afecta a una aplicación del usuario, ICMP deberá articular los mecanismos necesarios para comunicar a la aplicación el evento ocurrido.

Aunque este protocolo fué diseñado para detectar las incidencias que se producen en el transporte de un datagrama hacia el host destino, no todas ellas pueden ser detectadas. Entre estas causas se encuentra la pérdida de un datagrama que lleva un mensaje ICMP. En este punto, podríamos pensar que para solucionar este problema, esta pérdida podría ser notificada con otro mensaje ICMP. Más que solucionar el problema, lo estaríamos agravando cuando la razón de esa pérdida sea una congestión en la red. Por eso, NO SE PERMITE la notificación de mensajes ICMP causados por la pérdida de datagramas que lleven un mensaje ICMP. Otra norma general que impone este protocolo es que las notificaciones de error se envían SÓLAMENTE al host origen.

Campo	Tipo de mensaje
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo Request
11	Time exceeded
12	Parameter Problem
13	Timestamp Request

14	Timestamp Reply
15	Information Request
16	Informatión Reply
17	Address Mask Request
18	Address Mask Reply

Cada tipo de mensaje ICMP tiene su propio formato, aunque todos ellos comienzan con tres campos comunes, el resto puede variar en función del tipo de mensaje: El campo TIPO identifica el tipo de mensaje ICMP (ocupa 8 bits). En la Tabla anterior se muestran los distintos tipos de mensajes que contempla este protocolo. El campo CÓDIGO se usa para dar más información acerca del tipo de mensaje ICMP (8 bits). Y el último campo, contendrá el CHECKSUM de todo el mensaje ICMP (16 bits). El cálculo del checksum es el mismo que en IP, solo que en este caso cubre todo el mensaje ICMP.

En la tabla se muestran los distintos tipos de mensajes ICMP que considera el protocolo. De todos ellos, los más importantes son los seis primeros, que a continuación pasamos a describir brevemente:

• **Echo Request y Echo Reply:**
Son dos mensajes que se usan conjuntamente para determinar la alcanzabilidad de un host o una pasarela. Normalmente son utilizados por los hosts, de forma que estos pueden extraer información acerca del estado del host remoto, el retardo que introduce la red en la entrega de los mensajes y el porcentaje de mensajes perdidos.

• **Destino inalcanzable (Destination unreachable):**
Se trata de un mensaje que es generado por una pasarela cuando no puede encaminar un datagrama. Existen diferentes causas que provocan la emisión de este mensaje y que están codificadas en el campo código del mensaje ICMP. El mensaje es dirigido al host que ha enviado el datagrama, y en su interior se especifican los primeros 64 bits del datagrama que lo ha causado.

• **Source Quench:**
Es un mensaje que utilizan las routers para frenar el ritmo de inyección de mensajes en la red de un determinado host. Esta situación se produce cuando una pasarela se ve sobrecargada con la recepción de datagramas (una posible situación de congestión) teniendo que descartar algunos por falta de buffers. Cuando se produce esta situación, la pasarela envía un mensaje de este tipo al host origen del datagrama descartado, diciéndole que baje el ritmo de inyección de datagramas ya que en ese momento hay una situación temporal de congestión.

• **Cambio de ruta (redirect):**
Este mensaje es utilizado por una pasarela para indicar a un host de su red IP, un cambio en su tabla de encaminamiento, debido a la existencia de otra pasarela en la red que es más idónea que la que está utilizando actualmente.

• **Tiempo de vida agotado (Time Exceeded):**
Cuando una pasarela encamina un datagrama, una de sus tareas es decrementar en una unidad el campo TTL de la cabecera del mismo. Si tras la operación el campo vale "0", debe descartar el datagrama y enviar un mensaje ICMP de este tipo hacia el host origen.

Otras Fuentes de Información

A continuación se les presenta una lista de fuentes importantes de información de la resolución de problemas:

- Manuales de cursos que haya tomado
- Otros libros
- RFCs
- Páginas web
- Grupos de Noticias

ESCENARIOS DE RESOLUCION DE PROBLEMAS

Hay algunos problemas que usted va a enfrentar más a menudo que otros:

- Dificultad de poner hardware a funcionar correcta
- Problemas de resolución de nombre
- Conexiones intermitente entre los hosts
- Ninguna conexión entre ciertos host

En esta sección se incluyen procedimientos que usted puede seguir al tratar de solucionar problemas de averías problemas de la red. Discutiremos los siguientes problemas de red:

- Acabo de Instalar un Nuevo NIC, Pero No Puedo Interconectar
- Sólo Puedo Interconectar Usando Direcciones IP
- ¿Por Qué Son Algunos Host en Mi Red Inalcanzables?
- Puedo Interconectarme, Pero Pierdo 50 Por Ciento De Mis Paquetes
- ¿Por qué Consigo Conexiones De Red Intermitentes?

Acabo de Instalar un Nuevo NIC, Pero No Puedo Interconectar

Hay muchas razones para la ocurrencia de un problema tal como este, el más común implica fallas con el hardware y configuraciones de los módulos. Antes de involucrarnos en cuestiones demasiadas técnicas, uno debe hacer algún tipo de examinación básica para asegurarse de que el problema no es realmente un descuido. Si después de dicha examinación, se determina que el problema proviene de situaciones más generales, puede ser necesario hacer el ajuste a la configuración de la tarjeta.

Asegúrese de que Otros Medios de la Red Estén Trabajando Correctamente

Lo primero que uno debe de asegurarse cuando una nueva tarjeta de red no esté trabajando correctamente es si está funcionando correctamente el resto de los componentes de la red. No tiene ningún sentido pasar horas haciendo ajustes a la configuración del NIC cuando el problema podría ser fácilmente un cable malo de la red. Problemas similares podían también presentarse de un hub que está funcionando incorrectamente. Una solución a esta problematica es intentar conectar el cable en la NIC en una máquina que uno está seguro que trabaja correctamente. Si esa máquina experimenta problemas de la funcionalidad de la red similares a los de la máquina que contiene el nuevo NIC, entonces es cierto que el problema yace en otra parte en la red. Si usted no tiene otra máquina disponible para realizar una prueba de esta naturaleza, intente cambiar cualesquiera cables o dispositivo de red en su red.

Asegúrese de Que el NIC Esté Asentado Correctamente

No importa que en particular que también este configurada una tarjeta si no está asentada bien físicamente. Por lo General, esto significa que la tarjeta no está dentro del slot de la placa madre. Cuando esto sucede, el sistema no puede comunicarse con el NIC. Cuando la tarjeta está instalada, usted debe cerciorarse de que la tarjeta ajuste con seguridad en lugar. Uno nunca debe forzar una tarjeta a que ajuste; sin embargo, una fuerza moderada es a veces necesaria.

Si nada de esto funciona y aún esta sospechoso, a este punto, el usuario puede también intentar instalar el NIC en otro slot o ranura ya que puede ser que la original este en falla.

Compruebe las Configuraciones de Red

Si la tarjeta está bien instalada o probada en otro ordenador y funciona debemos pasar a usar un programa de la configuración, tal como netconfig o ifconfig, para asegurarnos de que los parámetros de red estén establecidos correctamente en el sistema. Estas utilidades de configuración de red, a través de una serie de preguntas, le permiten al usuario incorporar tal información como el nombre de host, la dirección IP de la máquina, el nombre del servidor y la pasarela (gateway) por defecto. Asegúrese de que toda esta infor-

mación se digite correctamente. Si esta información no está disponible, deberá obttenerla con el ISP o el administrador del sistema.

Compruebe Las Configuraciones De Hardware

Si se ha comprobado que no hay problema con el resto de la red y que el NIC está asentado correctamente, usted puede asumir con relativa confianza que la solución ya está cerca haciendo algunos ajustes a la configuración del NIC. Antes de hacer estos ajustes, sin embargo, compruebe para asegurarse de que la tarjeta esté siendo reconocida por la máquina. La manera más fácil de hacer esto es mirando los mensajes almacenados en el kernel ring buffer. En cada arranque del sistema, el kernel ring buffer registra cada mensaje del sistema. Por lo tanto, este almacenador intermediario (buffer) debe contener la información sobre el NIC, si es que fué detectado a la hora de cargar. El comando dmesg está específicamente diseñado para exhibir la información en el kernel ring buffer. Una importante observación es que una vez se llene este almacenador intermediario, continúa agregando los mensajes más recientes y eliminando los más viejos. Por esta razón, el usuario debe utilizar el dmesg cuanto antes después del arranque.

Al revisar la salida del dmesg, busque las palabras claves como eth0, marca del NIC, o cualquier otra información que se pueda relacionar con la tarjetas de la red. En esta etapa, la principal preocupación es ver si la información está presente más que intentar de interpretarla. Si está presente, el usuario sabe que la tarjeta está siendo reconocida por el sistema. Si no está presente, la tarjeta no se ha detectado. El paso siguiente es agregar el módulo apropiado para el NIC. Para este propósito, es mejor utilizar una utilidad de la configuración, tal como netconfig, si es que está disponible. Las utilidades de la configuración permiten que usted realice fácilmente cambios a las configuraciones de hardware. Si el usuario aún experimenta problemas después de intentar una utilidad de configuración, necesitará aplicar alguna de la configuración manual que discutiremis más adelante en esta sección.

Si no hay una utilidad de configuración disponible en el sistema, los módulos se pueden cargar usando el comando del modprobe. Si se asume que depmod -a ha sido ejecutado desde el último build (etapa de compilación del kernel que prepara los módulos para luego poder ser linkeados) y la instalación más reciente del kernel completo o de un módulo nuevo, el modprobe consultará el árbol de dependencia en el archivo modules.conf (o en versiones anteriores, conf.modules) para confirmar qué serie de comandos necesita para cargar manualmente un módulo en el kernel. El modprobe también permite que el usuario especifique las direcciones IRQ y de I/O que se deben utilizarse para su hardware. Por ejemplo, para cargar un módulo para un NIC NE2000 de nombre ne2 en la dirección IRQ 15 y una dirección base de I/O 0x380, usted ejecutaría el modprobe con el sintaxis siguiente:

modprobe ne2 irq=15 ioprot=0x380

Utilice el modprobe para cargar el módulo necesitado por su NIC usando el mismo sintaxis y los parámetros específicos a su tarjeta. Si no estea seguro de los ajustes de las direcciones IRQ y I/O, lo discutiremos más adelante en esta sección. Después de reiniciar el sistema, compruebe los mensajes de sistema usando el comando dmesg. Si todavía no hay información pertinente a su NIC, puede ser necesario que intente cargar un módulo diferente. Si aparece la información, está claro que el módulo funcionó y que fué cargado, y el sistema reconoce que el NIC está instalado.

Compruebe la Dirección de IRQ y la Dirección Base de I/O

Si el sistema reconoce el NIC, pero aún no hay comunicación con la red, puede ser que el NIC esté utilizando una dirección de IRQ o una dirección base de I/O que esté en conflicto con la de otro dispositivo. Antes de intentar configurar estos aspectos del NIC, usted debe entender que significan estos términos y como ellos trabajan.

IRQ son la siglas para la petición de la interrupción. Un ajuste de IRQ le dice a una computadora dónde

buscar señales de interrupción de un dispositivo instalado. Por ejemplo, cuando la tarjeta de red realiza una operación, enviará una señal al computadora para que esta sepa su estado. La computadora entonces sabrá como asignar prioridades a sus procesos. Si más de un dispositivo es asignado una sola dirección IRQ, el computador se confundirá en de donde proceden las señales. Por lo tanto, usted no puede asignar a múltiples dispositivos la misma dirreción IRQ.

Una lista de qué direcciones IRQ están en uso por un sistema y los dispositivos que las usan está situada en el archivo /proc/interrupts. Un usuario puede deducir al ver este archivo si el NIC está presente en cualquier IRQ en particular o uno puede buscar IRQs disponible para utilizar al hacer las especificaciones para el módulo del NIC. Aquí le mostramos un archivo /proc/interrupts:

```
k1k1@kikla:~$ cat /proc/interrupts
          CPU0
  0:   5473807        XT-PIC  timer
  1:      3982        XT-PIC  i8042
  2:         0        XT-PIC  cascade
  3:     38394        XT-PIC  orinoco_cs
  5:        52        XT-PIC  ALI 5451
  7:        11        XT-PIC  parport0
  8:         1        XT-PIC  rtc
  9:     15194        XT-PIC  acpi
 10:       101        XT-PIC  eth0
 11:         7        XT-PIC  yenta
 12:    298498        XT-PIC  i8042
 14:     24523        XT-PIC  ide0
 15:     49204        XT-PIC  ide1
NMI:         0
LOC:         0
ERR:         0
MIS:         0
```

Aquí, la primera columna es la dirección de IRQ, y la última columna indica qué dispositivo está usando una dirección particular.

Una dirección de I/O específica el rango de direcciones en memoria que puede ser utilizada por un dispositivo instalado a un sistema. Como un ajuste de IRQ, un dispositivo que es asignado una dirección I/O no puede estar en conflicto con otros dispositivos o el dispositivo no funcionará, y en algunos casos, el sistema cesará de funcionar en su totalidad. Para ver una lista de las direcciones de I/O en uso y los dispositivos que los están utilizando, visualice el archivo /proc/ioports. Al igual que /proc/interrupts, este archivo es útil para determinar si el NIC está trabajando o para encontrar un ajuste disponible si no está trabajando. Debajo está la muestra de un archivo /proc/ioports:

```
k1k1@kikla:~$ cat /proc/ioports
0000-001f : dma1
0020-0021 : pic1
0040-005f : timer
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0100-013f : pcmcia_socket0
0170-0177 : ide1
01f0-01f7 : ide0
```

02f8-02ff : serial
0370-0371 : pnp 00:00
0376-0376 : ide1
0378-037a : parport0

Aquí uno puede ver un listado de las direcciones I/O seguida inmediatamente por los nombres de los dispositivos conectados.

Si, después de realizar los procedimientos descritos en esta sección, el NIC aún no funciona como debe, contacte al vendedor o al fabricante del NIC para la ayuda adicional.

Puedo Conectarme Solamente Con Direcciones IP

Este problema no involucra los cables de la red puesto que si hay la conexión usando la Dirección IP del servidor. El único problema aquí está en resolver los nombres simbólicos a las direcciones IP, que se asocian en DNS. Los siguientes son algunos posibles errores que pueden ocurrir y la solución sugerida.

El Archivo de Configuración /etc/host.conf Pueden que Contenga Errores de Sintaxis

El administrador de sistema necesita estar seguro que el orden en este archivo es la exacta en la cual resolver nombres de host a direcciones IP. Si se incluyen opciones en el archivo/etc/host.conf, deben ocurrir en líneas separadas.

order hosts nis bind
multi on

En este ejemplo, la opción "order" le dice al sistema que para determinar la información requerida primero revise el archivo /etc/hosts, después uso NIS para la resolución de nombres y finalmente use el BIND para consultar un servidor del DNS. La opción "multi on" le dice al sistema que se utilizarán direcciones múltiples.

El Nombre Dominio y la Dirección IP del Servidor de Nombre se Pudieron Haber Establecido Incorrectamente

El administrador de sistema necesita revisar el archivo /etc/resolv.conf en el host local para verificar que el sintaxis está correcto. Si el sintaxis no está correcto en una línea en archivo /etc/resolv.conf, entonces el DNS simplemente ignora esa línea. Esto podría causar que el Servidor de Nombre en particular sea inasesible al DNS.

Puede Ser Que el Servidor de Nombre No Responda

El administrador de sistema debe tener por lo menos dos Servidores de Nombres en el archivo /etc/resolv.conf para asegurarse de que el segundo Servidor de Nombres se pueda utilizar para resolver nombres de host en el acontecimiento que el primer servidor de nombres no esté respondiendo.

Puede Que el DNS en el Servidor de Nombres No Se Este Ejecutando

Primero, el administrador del sistema necesita comprobar si el archivo de boot (cargar) del named o si está presente el archivo /etc/named.boot. El demonio named debe estar ejecutándose en el Servidor de Nombres para que el DNS pueda estar operacional.

El Servidor del DNS No está funcionando Correctamente

La manera más fácil de confirmar que el servidor del DNS está funcionando correctamente es usando el comando del nslookup. El administrador de sistema necesita cerciorarse de que el comando nslookup asigna el servidor de nombre por defecto a la dirección IP. La salida del comando nslookup es mostrado a continuación para el servidor de nombre ficticio "ns.codigolibre.org":

$nslookup

Default Server: ns.codigolibre.org

```
Address:        10.0.0.1
    >set all
Default server:         ns.codigolibre.org
Address:        10.0.0.1

Set options:
  novc                  nodebug         nod2
  search                recurse
  timeout = 0           retry = 2       port = 53
  querytype = A         class = IN
  srchlist = sede.codigolibre.org"
```

El administrador de sistema puede utilizar el comando "set all" después del prompt para obtener el dominio por defecto, lista de búsqueda, el tipo de consulta, y otras opciones además de la dirección del servidor de nombre, según lo mostrado en el ejemplo anterior.

Si el administrador de sistema no puede conseguir esta salida, él o ella sabe que hay un problema con el servidor de nombre y verifica otra vez el archivo /etc/resolv.conf.

Una vez que el nslookup esta funcionando y dá la salida deseada, el administrador de sistema no debe tener dificultad al conectarse con el servidor usando nombres de host.

Para los host más comúnmente usados, el archivo /etc/hosts puede ser utilizado para la traducción de los nombres del dominio a las direcciones de IP y de los nombres cortos y viceversa.

¿Por qué son inalcanzables algunos host en mi red?

Este problema puede venir en varias formas extendiendose desde influencias globales a las locales. Las posibilidades son relativamente fáciles de apuntar una vez que se entienden cuales herramientas serán necesitadas diagnosticar el problema. Este topico de resolución de problemas se divide en tres secciones. Primero, las influencias globales, las cuales suelen ser la fuente del problema la mayor parte del tiempo, son discutidos desde la perspectiva del mundo exterior (nonuser). En segundo lugar, el factor local que afecta la conexión de una red con un host. La influencia local es basada en la localización inmediata. Las causas obvias y básicas del fallo de la red se cubren en esta sección. Finalmente, la tercera sección proporciona diagnóstico si las soluciones proporcionadas fallan. A continuación se muestra un panorama específico.

La idea de una LAN es conectarse con un host, un Web site, o una computadora usando una red 10BaseT. La conexión de red va directo a un hub, que, alternadamente, entra en un zócalo de pared RJ-45 (la topología de la red no es completamente reelevante para esta parte). La estacion de trabajo se sitúa en una pequeña oficina con varias otras computadoras que funcionan con una interface 10BaseT.

Influencias Globales

Las influencias globales son factores que los usuarios no pueden controlar. Los problemas pueden implicar:

- El host debe apagarse en el proceso de reinicio del sistema.

  Los administradores de sistema necesitan actualizar constantemente sus computadoras con un mejor hardware y software. Este proceso colocará generalmente al ordenador fuera de línea temporalmente. En este caso, no tiene nada hacer con la configuración del 10BaseT. También, un host puede tener períodos donde

se está comunicando con varias computadoras, asi, de esta manera la red esta ocupada. Por esta razón, una conexión pudo fallar.

Solución

La solución más fácil es esperar e intentar más adelante. Una herramienta de diagnóstico simple y con muchos recursos que pueden ser usados es el comando ping. Corriendo este comando con la dirección IP, un usuario puede descubrir rápidamente si el ordenador está corrriendo basado en losreportes de paquetes perdidos.

- Los operadores de red pudieron no haber colocado o haber colocado incorrectamente su dominio con el Domain Name System (DNS).

Esto evitaría que todos los usuarios en el Internet tuvieran acceso al Web site o al host.

Solución

Utilice un newsgroup del usuario para encontrar información sobre este host o para hacer una búsqueda más detallada para el nombre/URL del host con un motor de búsqueda. Esto puede revelar la información adicional al dominio apropiado.

- Otro ejemplo de no poder tener acceso a un host es cuando ha ocurrido una interrupción física en la conexión.

Se ha interrumpido el cable que víncula el host particular al Internet. Una sección del cable se pudo haber desconectado o peor aún cortada, preveniendo al cables de enviar y recibir los paquetes de la transmisión.

Solución

La única opción es esperar hasta que se reparan los cables.

- Al comunicarse por el Internet, se necesitan un conjunto de protocolos para que asi se puedan comunicar las computadoras entre si. Pasarelas son utilizadas para transferir información de un protocolo a otro.

Diferentes redes pueden usar diferentes protocolos tal como TCP/IP o IPX/SPX. Siempre y cuando estos protocolos pasan a través de una pasarela que víncule los dos protocolos, la comunicación prevalecerá. Si la pasarela sale fuera de línea, entonces reinará la posibilidad de ninguna comunicación al mundo exterior.

Solución

Si la pasarela está fuera, esperar que esta sea corregida es la única solución.

Influencias Locales

Las influencias locales implican factores que el usuario tiene poco control, incluyendo:

- Una conexión física de un cable de UTP a una tarjeta de Ethernet se desconecta o se está conectando mal.

La solución es comprobar si la tarjeta de Ethernet está trabajando correctamente y si está funcionando con los ajustes correctos. Otra cosa es que los LED en el NIC deben estar encendiendo si hay actividad.

- El medio de conexión (es decir, twisted pair) se ha dañado entre el terminal del usuario y el hub.

Esto ocurre a menudo cuando los cables no se construyen correctamente. Cuando los cables están a la interperie, la posibilidad de que se mal traten y de que los alambres se rompan dentro del protector de revestimiento aumenta. La solución es absolutamente simple: los alambres deben ser colocados donde la gente no puedan caminar sobre ellos, rodar sillas sobre ellos, o colocar cualquier cosas sobre ellos.

Diagnóstico

Si las soluciones presentadas anteriormente no resuelven el problema, pruebe con las siguientes:

- Asegúrese de escribir la URL correctamente. Por ejemplo puede ser que escriba la letra x por la letra c al escribir "com".
- ¿Es este host legítimo? Compruebe para ver si el host tiene una dirección inválida comprobando con el DNS.
- Ejecute un ping con la Dirección IP del host para ver si hay una respuesta. El comando es así:

$ ping 10.0.0.105 -c 20

- Si tiene problemas al conectar un ordenador en el hogar, compruebe el cuarto punto de Influencias Globales y la primera bajo Influencias Locales. Los protocolos se pueden pasar por alto al instalar el sistema operativo.

Puedo Salir, Pero Pierdo 50 Por Ciento De Mis Paquetes

Un paquete es la estructura estándar que se utiliza para encapsular los datos que son enviados sobre una red TCP/IP. Cuando una computadora envía paquetes a una computadora en el mismo LAN, la computadora destinataria recibe los paquetes directamente. Cuando las computadoras no están en la misma red física, los paquetes se pasan a través de dispositivos de red hasta que alcanzan el sistema remoto. Para proporcionar servicio confiable, el TCP utiliza un protocolo basado en reconocimiento. Siempre que se envíe un paquete, la computadora de recepción envía un mensaje que indica que recibieron el paquete y que más paquetes deben ser enviados. Se utilizan números de serie para mantener los paquetes orden. El TCP normalmente envía un número de paquetes juntos y espera un solo reconocimiento por esos paquetes. Si el reconocimiento no se recibe, la computadora que envía vuelve a enviar esos paquetes y espera otra respuesta. Si un reconocimiento no se recibe otra vez, los paquetes se consideran perdidos.

Los ruido en la líneas y relámpago son dos tipos de interferencia que pueden contribuir a la pérdida de paquetes; sin embargo, interferencia física no es una causa significativa de la pérdida de paquetes. Normalmene los paquetes son perdidos debido a congestión en la red. Pueden ser retrasados dentro de bucles de enrutamiento, ser recibidos fuera de orden o desechados debido a desbordamiento. Durante el volumen de tráfico alto, puede ser que un sistema reciba más paquetes que los que puede procesar. Una vez que los buffers esten llenos, un desbordamiento de los buffers puede ocurrir, lo que significa que desecha cualquier paquete nuevo hasta que el el buffer ya no este lleno. El desbordamiento del buffer es la razón más común de la pérdida de paquetes y puede ocurrir en el dispositivo físico, el driver del dispositivo, o la puesta en práctica del protocolo. Siempre que un paquete sea desechado debido al desbordamiento del buffer, la computadora de recepción no podrá reconocer el sistema de paquetes. La computadora que envía medirá el tiempo de entrega, declarará los paquetes perdidos, y enviará ese conjunto de paquetes otra vez. Alta pérdida de paquetes aumenta la congestión en la red y causa perdida de efectividad en la red.

Determinación la fuente de la pérdida de paquetes puede ayudar en aislar problemas dentro de una red. La pérdida de paquetes se puede detectar con un número de utilitarios estándares.

El Utilitario ping

Ping es un programa utilitario básico usado para comprobar si hay una respuesta en la red de un dispositivo. Ese envía una petición de eco ICMP al sistema remoto y espera por una respuesta por eco del ICMP. Los mensajes del ICMP funcionan en la capa de red, así que ping recibirá una respuesta de una máquina remota sin importar la plataforma que la máquina está funcionando. El ping le mostrará el tiempo de reacción al host, el TTL del paquete y un número de secuencia. El campo TTL realmente no es un campo de tiempo, sino una cuenta de los saltos que disminuye de un número base, normalmente 255, cada vez que el paquete encuentra un enrutador o una pasarela en su trayectoria. Una vez el contador llega a 0, el paquete es desechado.

Una vez el comando ping se termina (usando la combinación de teclas CONTROL+C), se efectuá un resumen de las respuestas y el porcentaje de la pérdida de paquetes se exhibe. Así, que podemos usar el ping para demostrar si está ocurriendo una alta pérdida de paquetes, ping tiene tres opciones las cuales pueden ser útil:

-v Cambia al modo verbose. Imprimen todos los mensajes del ICMP, no solamente respuestas del eco.

-c Esta opción es la cuenta con un argumento numérico, ping enviará solamente la cantidad de paquetes dados en el argumento.

-f Esto es el ping de inundación, una opción que pone rápidamente mucha carga en la red. El ping puede rápida y repetidamente efectuar ping a otros hosts (esta opción requiere privilegios del superuser).

Los problemas, tales como el reporte falso de porcentaje de la pérdida de paquetes, especialmente 50 por ciento, existe con algunos clientes de ping más viejos. Para solucionar este problema, simplemente debe actualizar el paquete de las herramientas de red “net-tools” a la última versión disponible para su distro.

El Utilitario netstat

El comando “netstat -i” mostrará la información sobre las interfaces actualmente configuradas en la máquina local. La salida puede exhibir los paquetes pérdidos o los problemas de la máquina local con los paquetes duplicados o dañados. La siguiente salida se abrevia para demostrar las columnas importantes:

```
# netstat -i
Kernel Interface table
Iface  MTU   Met  RX-OK  RX-ERR RX-DRP RX-OVR  TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1500  0    0      0      0      0       109   0      0      0      BMU
eth1   1500  0    0      0      0      0       0     102    102    0      BMRU
eth2   1500  0    4575   0      0      0       2572  2      0      0      BMRU
lo     16436 0    1553   0      0      0       1553  0      0      0      LRU
sit0   1480  0    0      0      0      0       0     100    0      0      ORU
```

Las columnas RX son para los paquetes recibidos y las columnas TX son para los paquetes transmitidos. En el ejemplo anterior, ningunos paquetes dañados (ERR), los paquetes caídos (DRP) o las pérdidas debido al overrun (OVR) han ocurrido. Si ocurren paquetes dañados, es probable que es un problema de hardware exis en la red. Si los desbordamientos no están ocurriendo en la máquina local, el problema puede estar ocurriendo en otra parte en la red.

El Utilitario traceroute

La herramienta traceroute se puede utilizar para aislar los problemas de enrutadores en una red. El traceroute utiliza un mensajes pequeño UDP para buscar la pista y mapear la ruta que el paquete llevaría para arrivar a un host y el valor aproximado del tiempo del viaje. Usando la información obtenida de traceroute, uno puede continuar buscando el problema efectuando ping a los enrutadores en toda la ruta, lo que puede aislar el problema a un router en específico. La salida de traceroute mostrará nombres de host o si usa-

mos la opción -n, sólo mostrará las direcciones IP. Típicamente el comando traceroute se ejecuta de la siguiente forma:

/usr/sbin/traceroute

¿Por Qué Logro Conexiones De Red Intermitentes?

Este problema se presenta muy probablemente por el hecho de que alguien está compartiendo una misma Dirección IP. Si las Direcciones IP de los usuarios se asignan estáticamente, entonces debemos cerciorarnos de utilizar la dirección que se nos fué asignada. Ingresar al sistema bajo una Dirección IP incorrecta causará problemas para el usuario y con la persona que está compartiendo la dirección.

¿Cómo Puedo Detectar Si Alguien Está Utilizando Mi Dirección IP?

Simplemente apague su computadora, inicie una sesión en otra máquina y efectué un ping al IP de su ordenador asignado (o sea al que acabas de apagar). Si se responde el ping, entonces alguien está usando esa dirección.

¿Qué Debo Hacer Sobre Esto Situación?

Lo primero que puede hacer es ejecutar un traceroute al otro host. Con esto determinará si el culpable está en la misma subnet que usted. Si no lo está, entonces el traceroute determinará la pasarela más cercana a la máquina que está usando el IP. Esta información asistirá al administrador del sistema en intentar corregir el problema. Las instrucciones para usar el comando traceroute la encontrará en las páginas man de su GNU/Linux.

Si el que está usando el IP está en la misma subnet, entonces podemos usar el comando arp para determinar la Dirección MAC ADDRESS de la otra máquina. Ya habiendo efectuado el traceroute, el otro host ahora ya está en la tabla del ARP de la máquina local. Usando el “arp -a” para enumerar el contenido de la tabla del ARP y podemos comprobar la salida para saber el MAC ADDRESS del culpable. Éste es un número que identifica de la tarjeta de red individual de la máquina y dependiendo de la topología de la red, podría dar al administrador bastante información para encontrar el terminal usando la Dirección IP ajena. Claro está, esto sólo trabajará si el usuario y el host usando la IP indebidamente están en la misma red.

MONITOREAR EL RENDIMIENTO

Mientras mejor usted conoce su red, más fácil es encontrar los problemas que se le presenten en ella. El monitoreo y supervisión regular del rendimiento de su red le pueden proporcionar la detección preventiva de los problema que pueden surgir y agilizar el proceso de los esfuerzos para la resolución de los problemas futuros. En Esta sección se describen los elementos del monitoreo y la supervisión del rendimiento en como se aplican al proceso de resolución de problemas.

Los siguientes tópicos serán cubiertos en esta sección:
- Factores del Rendimiento
- Identificar la Degradación del Rendimiento
- Ambiente del Sistema
- Ambiente de la Red
- Aplicaciones Cliente/Servidor

Factores del Funcionamiento

La identificación oportuna de entaponamientos es un factor dominante para mantener un constante nivel de rendimiento. La mayoría de los usuarios no son tolerantes de las aplicaciones cliente/servidor que su rendimiento sea inconsitente. Asi que el primer objetivo es obtener consistencia tanto en el fun-

cionamiento de las aplicaciones como el rendimiento. Claro se supone que para detectar inconsistencias, un administrador debe tener un parámetro contra el cual medir, la llamaremos la línea base del rendimiento que normal la red.

Línea Base

La línea base es grabar la actividad de la red en un momento de rendimiento satisfactorio para que nos sirva de parámetro para comparar contra futura actividad de la red. Las líneas base deben ser registradas cuando una red está funcionando correctamente. Si se introducen problemas a la red, el nuevo comportamiento de la red se puede comparar contra la línea base. Las líneas base se pueden utilizar para determinar entaponamiento, para identificar patrones tráfico denso y para analizar el uso diario de la red y los patrones de los protocolos usados en la red.

Identificar la Degradación del Rendimiento

Una aplicacióncliente/servidor de red que anteriormente funcionaba correctamente pero que recientemente se ha tornado inconsistente en su comportamiento típicamente indica que uno o más de sus recursos están siendo utilizados a sus límites.

Sistema

Puede ser que la aplicación este esperando en algunos recursos del sistema como por ejemplo:

- El CPU en el nodo parte del Cliente en el cual se ejecuta una parte de la aplicación.
- El CPU en el nodo parte del Servidor en el cual se ejecuta una parte de la aplicación.
- El CPU en el nodo del Servidor al que la aplicación necesita tener acceso (por ejemplo, un aplicación de tres capas que accesa una base de datos en el servidor).
- Memoria en el nodo del Cliente que la aplicación se ejecuta.
- Memoria en el nodo del Servidor que parte de la aplicación se ejecuta.
- Memoria en el nodo del Servidor que la aplicación necesita tener acceso

Network

Los recursos de la red que afectan el rendimiento de una aplicación incluyen:

- La LAN en la que el nodo Cliente está configurado
- La LAN (sistema operativo o aplicación) en la que el nodo del servidor está configurado.
- La LAN en la que la aplicación nodo del servidor está configurada.
- La WAN que conecta el nodo Cliente con el nodo Servidor (dependiendo de donde las aplicaciones se ejecutan con respecto a la red)
- Arquitectura y configuración de los dispositivos de comunicación.
- Protocolos usados entre el cliente y el servidor de la aplicación.

Aplicaciones Cliente/Servidor

Determinar el uso de los recursos del sistema y de red críticos no es suficiente. El diseño y la arquitectura de las aplicaciones cliente/servidor y el servidor de aplicacion es configurado y administrado son factores extremadamente importantes en análisis de funcionamiento. El funcionamiento de la aplicacion debe ser pobre por que la aplicacion de la arquitectura fueron diseñados mal. Varios factores deben ser considerados:

- ¿Es la mayoría de los datos accesados por los usuarios finales de solo lectura y lectura/escritura?•
- ¿Qué opciones el servidor de aplicaciones proporciona para maximizar funcionamiento en un sistema de un solo procesador?
- ¿Qué opciones el servidor aplicaciones proporciona para maximizar el funcionamiento en un sistema de múltiples procesadores?

- Es posible que los procesos del servidor de aplicaciones puedan estar ocupados pero el sistema esta siendo usado. ¿Cómo puede usted mejorar el uso del sistema con los procesos del servidor de aplicaciones?
- ¿Cuál es la arquitectura del servidor de aplicaciones? ¿Se centraliza o se descentraliza? ¿Por qué? ¿Es este arreglo constante con la arquitectura de red?

En el lado de la aplicacion, la arquitectura de aplicacion y cómo este código puede significativamente afectar la utilización de los recursos del sistema y de la red. ¿Por ejemplo, cómo es implementado el control de la version? ¿Hay un mecanismo que proporcione la información sobre el sistema de la carga y la red como resultado de los cambios realizados en la aplicacion?

Establecer Pautas

Para solucionar el problema de inconsistencia y pobre funcionamiento de la aplicacion, usted debe establecer pautas específicas en las áreas siguientes:

- Ambientes del Sistema
- Ambientes de la Red
- Aplicaciones Cliente/Servidores

Ambiente de Sistema

El ambiente de sistema incluye las dos áreas siguientes:
- Sistema del Hardware
- Sistema Operativo

Sistema del Hardware

Las aplicaciones cliente/servidor se ejecuta por lo menos en dos sistemas pero puede ser que funcione en más. Típicamente, tres sistemas críticos son asociados:

- PC del usuario (el interfaz GUI).
- El sistema que funciona como servidor de archivo para la PC del usuario. El ejecutable de la aplicacion que reside en el servidor de archivo del sistema.
- El host que esta configurado como el servidor.

La configuración de todos los sistemas debe estar claramente definida. Los elementos importantes de la configuración son los siguientes:

- **Procesador**
  - Tipo de CPU (Pentium de Intel, alfa Digital, sol SPARC)
  - Número de CPUs

  ¿Cuál es el número máximo de CPUs que puede ser instalado en este sistema?

- **Memoria**
  - **Memoria Real**

  ¿Cuánta memoria está instalada en el sistema? ¿Cuál es la cantidad máxima de memoria que podría ser instalada?
  - **Memoria Virtual**

  ¿Monto de memorias secundarias que han sido configuradas en el sistema operativo?
  ¿Cuál es la cantidad máxima de memoria que el sistema opetativo soporta?

- **Interfaz de la red**
  - ¿Qué tipo de interfaz de la red ha sido configurada en el sistema?
  - ¿Están la interfaz y el sistema operativo optimizados para transmitir la Unidad Máxima de Transmisión (MTU) definido por el protocolo en uso?

• ¿Están la interfaz de la red y el sistema operativo optimizados para aprovecharse del bús del sistema? ¿Por ejemplo, es el USB preferible a ISA o al PCI?

• **Disco**

• ¿Qué tipo de controladora y manejadora están instalado en el sistema?

En la siguiente tabla listamos los elementos del sistema y los factores que un administrador de la red debe tomar en consideración.

Sistema Operativo

Usted también debe determinar e indicar exactamente cuales componentes de las aplicaciones funcionan en sistemas Cliente/Servidor y cómo estos funcionan. Esta sección esplicara las herramientas para monitorear las herramienta para Linux.

En los sistemas GNU/Linux, usted puede utilizar los siguientes comandos para proporcionar información de como el sistema esta funcionando:

• vmstart
• ps
• uptime

El Comando vmstat

El comando vmstat (estadística de la memoria virtual) proporciona información de la memoria virtual, el acceso del disco, y el uso de la CPU. Los resultados proporcionados por este comando son promediados a partir del tiempo de que el sistema fué inicializado. Para obtener la información sobre la actividad máxima del sistema (que puede indicar entaponamiento potenciales del sistema), especifique un intervalo como un argummento del comando del vmstat.

Elementos Del Sistema	Consideraciones del Administrador de la Red
Configuración del Sistema Cliente	Determinar el procesador en el sistema cliente: Intel, Spac, etc Determinar la velocidad del reloj: 233 MHZ, etc.
Configuración del Sistema Operativo Servidor	Determinar el procesador en el sistema operativo servidor: Intel, Spac, etc Determinar la velocidad del reloj: 233 MHZ, etc.
Configuración del Servidor de Applicación	Determinar el procesador en el servidor de aplicaciones del sistema: Intel, Spac, etc Determinar la velocidad del reloj: 233 MHZ, etc.
Memoria Instalada/Memoria Maxima en el Sistem Cliente	Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras.
Memoria Instalada/Memoria Maxima en el Sistema Servidor	Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras.
Memoria Instalada/Memoria Maxima en el Servidor de Aplicaciones	Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras.
Memoria Virtual/Memoria Maxima Virtual en el Sistema Cliente	Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras.
Memoria Virtual/Memoria Maxima Virtual en el Sistema Servidor	Determinar que el sistema esta configurado optimamente para las aplicaciones cliente/servidor y otras.
Interface de Red del sistema Cliente	Determinar las interfaces: Ethernet, Token Ring, y/o FDDI
Interface de Red en el Sistema Operativo del Servidor	Determinar las interfaces: Ethernet, Token Ring, y/o FDDI
Interface de Red en el Servidor de Aplicaciones	Determinar las interfaces: Ethernet, Token Ring, y/o FDDI

Para la estadística de la CPU, especifique un intervalo de dos segundos. Para la estadística del disco, especifique un intervalo de 60 segundos. El comando vmstat duerme por el intervalo definido. Típicamente, si el tiempo libre de la CPU es mayor de 20 por ciento, implica que el sistema es límite de I/O o límite de la memoria. El tiempo libre de la CPU incluye la siguiente información sobre el tiempo de la CPU:

• La CPU no esta en uso por que no tiene nada hacer.
• La CPU está por esperando memoria.
• La CPU está esperando por una entrada/salida (I/O).

Examine la salida del comando vmstat en las columnas datos descrita en la siguiente tabla:

El Comando ps

Cuando el shell lanza un programa, se crea un nuevo proceso y se le asigna un número entero (PID) entre 1 y el 30,000, del cual se tiene la seguridad que va a ser unívoco mientras dure la sesión. Se puede verificar ejecutando el comando "ps", el cual muestra los procesos activos que tiene asociados a la terminal.

Un proceso que crea a otro se lo denomina proceso padre. El nuevo proceso, en este ámbito se le denomina proceso hijo. Este hereda casi la totalidad del entorno de su padre (variables, etc.), pero sólo puede modificar su entorno, y no el del padre.

La mayoría de las veces, un proceso padre se queda en espera de que el hijo termine, esto es lo que sucede cuando se lanza un comando, el proceso padre es el shell, que lanza un proceso hijo (el comando).

Cuando este comando acaba, el padre vuelve a tomar el control, y recibe un número entero donde recoge el código de retorno del hijo (0 = terminación sin errores, otro valor = aquí ha pasado algo).

Cada proceso posee también un número de "grupo de procesos". Procesos con el mismo número forman un solo grupo y cada terminal conectado en el sistema posee un solo grupo de procesos. (Comando ps -j) si uno de nuestros procesos no se halla en el grupo asociado al terminal, recibe el nombre de proceso en background (segundo plano).

Column	Descripción
r	La columna provee información en jobs que están disponibles. Si el siguiente número es alto, esto implica que el CPU es forzado a cambiar entre jobs disponibles. Este número debe indicar que el sistema esta destinado al CPU..
b	Esto provee innformacion en jobs durmiendo en una prioridad negativa, usualmente por un proceso que esta esperqando por disco, cinta, u otro recurso. Si este número es alto y el tiempo del CPU es alto, el sistema puede ser destinado a entrada y salida.
w	Esta columna específica el número de jobs ejecutados en los pasados 20 segundos y han sido sacado de la swap. Si este campo no es cero, el sistemas puede no tener suficiente memoria.

Es posible utilizar algunas variantes del comando "ps" para ver qué procesos se tiene en el equipo:

ps : muestra el número de proceso (PID), el terminal, el tiempo en ejecución y el comando. Sólo informa de nuestra sesión.
ps -e : de todas las sesiones.
ps -f : full listing: da los números del PID, del Ppid (padre), uso del procesador y tiempo de comienzo.
ps -j: da el PGID (número de grupo de los procesos - coincide normalmente con el padre de todos ellos).

El comando del ps, con opciones como aux y ef, proporciona información útil de los procesos que se ejecutan en el sistema. Provee información en:
• Uso del CPU por proceso (ver la columna de %CPU)

• Uso de la memoria por proceso (ver la columna de %MEM)
• Estado actual del proceso (ver la columna STAT)

El Comando uptime

El comando uptime proporciona información útil acerca de:
• El total del tiempo que el sistema ha estado ejecutándose.
• El número de usuarios ingresados en el sistema.
• Los promedios de las cargas de trabajos activos en el sistema en los previo 1, 5, y 15 minutos.

El kernel mantiene información sobre el promedio del número de trabajos activos en el sistema en los últimos 1, 5, y 15 minutos. El primer número proporciona información sobre la carga actual del CPU. Si el número es mayor a 4, el sistema puede ser limitado por CPU.

Otra herramienta disponible bajo licencia también GPL, es el comando top, este exhibe y actualiza la información sobre los quince primeros procesos del sistema.

Ambiente de Red

Desde el punto de vista de la red, existen diferencias significativas entre ejecutar una aplicación cliente/servidor sobre un LAN y ejecutar la misma apliación sobre un WAN. El impacto de la latencia de la red es más pronunciado en una WAN. Sobre todo esta diferencia es porque muchos segmentos de una WAN son de 56 Kbps o 1.5 Mbps, mientras que la mayoría de los segmentos de la LAN son de 10 Mbps ó 100 Mbps.

Los retrasos debidos a propagación y los introducidos por el procesamiento de los paquetes de los enrutadores afectaran el rendimiento de la aplicación client/server. Estos retrasos pueden ser más pronunciados en WANs que en LANs.

Factores del Rendimiento

En el área de red, el rendimiento de una aplicación puede ser afectada por lo siguiente:
- Pila del Protocolo
- Arquitectura del Enrutamiento
  - Protocolo de Enturamiento
  - Configuración del Enrutador
  - Saltos del Enrutador
- Ambiente WAN
- Ambiente LAN

Para resumir, la información siguiente es esencial para determinar si la red está entaponada:
- Uso promedio de los segmentos de la LAN que conectan el cliente y servidor.
- El Uso pico de los sistemas clave tales como nodo del cliente, sistema operativo del servidor y nodo del servidor de aplicaciones.
- Distribución de la trama en segmentos claves del LAN y la WAN. Use esta información para determinar si la aplicación o el servidor de aplicaciones tiene cualquier parámetro u opciones que afecten tamaño del paquete.
- Los tipos de protocolos en segmentos claves del LAN y la WAN. ¿Cuáles protocolos ponen mayor carga en la red? ¿Hay una manera de optimizar la carga de la red examinando donde los sistemas están situados en la red?

Aplicaciones Cliente/Servidor

El desarrollo y la implementación de aplicaciones cliente/servidor sobre una red a menudo requieren de un administrador de red y un desarrollador de aplicaciones. Para determinar cómo las aplicaciones cliente/servidor se ejecutan, usted debe confrontar las siguientes áreas:

• Arquitectura de las aplicaciones en términos del sistemas y de las redes
• Arquitectura de las aplicaciones en términos de los módulos (pantallas, rutinas)
• Control de la versión
• Prueba

ADMINISTRACIÓN DE LA RED Y SNMP

La necesidad imperante de la administración de la red y las herramientas como el SNMP, se ha presentado debido a la complejidad de las redes de hoy. A menudo, la solución de problemas puede ser muy mejorada usando el SNMP (Simple Network Management Protocol).

A medidas que las redes crecen en tamaño y se tornan más complejas, la tarea de manejar los sistemas en una red se dificultan más. En los últimos años, un número de productos de la administración de la red han aparecido basados en un protocolo del suite del Internet conocido como SNMP. Esta sección nos introduce a la administración de la red usando el protocolo del SNMP.

Los siguientes tópicos son discutidos en esta sección:
• La Necesidad de la Administración de la Red
• Utilitarios Tradicionales de la Administración de la Red
• Administración Moderna de la Red
• Componentes de SNMP
• Estructura de la Información de Administración (SMI)
• Administración la Información Base (MIB) Espacio de Nombre del Objeto
• SNMP
• Otros Protocolos de la Admininstración de la Red

La Necesidad de la Administración de la Red

Causado por el crecimiento de la redes y debido a este crecimiento la dificultadad de sus administración de estas redes nació la motivación para el desarrollo de una herramienta estandarizada de administración de la red. El dianóstico de lsos problemas se torno más y más dificil. La necesidad de herramientas de administración de red TCP/IP se convirtió pronunciada a mediado de la decada de los años ochenta a medida que estos problemas comenzaron a utilizar más recursos de muchas compañías. Había una necesidad obvia, para un conjunto de reglas estandardizado que gobernaran la identificación y que suplan acciones automatizadas para las diversas situaciones comunes que se presentan en la red.

Las recomendaciones para un protocolo estándar en el año 1988 (RFC 1052) incluyeron:
• Que el SNMP es adoptado como un estándar internacional.
• Que el SNMP sea extendido y mejorado continuamente.

En el 1988, un protocolo llamado SNMP ya existiá pero no fué estandardizado y una variedad de otros protocolos también se encontraban en uso.

El RFC 1052 resumió la decisión del IAB (Internet Arquitecture Board) para adoptar el SNMP como un estándar en todo el Internet y para mejorar su especificaciones y para garantizar su mejoramiento continuado. La IAB también decidió que los protocolos OSI (CMIP/CMOT) sean adoptados en última instancia como

el reemplazo del SNMP. Sin embargo, como con algunos otros protocolos de la OSI, éstos han caído por el wayside, y el SNMP ahora es el estándar.

Utilitarios Tradicionales de la Administración de la Red

En el pasado las herramientas tradicionales de la administración de la red tendian a ser demasiado propietarias y dispersas para la solución de problemas y la prevención de las fallas que ocurren día a día en redes heterogéneas. Se necesitaba una herramienta específicas para cada trabajo y no existían la disponibilidad de soluciones integrada. Por lo tanto, las herramientas tradicionales de administración de red no eran lo suficientemente robustas para las redes grandes.

En las redes pequeñas, el conjunto básico de utilidades que provee normalmente los paquetes de software TCP/IP, es suficiente para administrarlas . Ya hemos mencionado los utilitarios principales del conjunto que se proveé con el paquete TCP/IP:

ifconfig

Se utiliza para comprobar o actualizar los parámetros de una interface de red en el sistema local. Usando el ifconfig, usted puede cambiar la Dirección IP de un interfaz, su mascara de subred o su dirección de broadcast (difusión). Usted puede detener un interfaz de acepta cualquier tráfico (apagarla) o reiniciar un interfaz (encenderla).

ping

Se utiliza este comando para probar si un sistema en particular se puede contactar desde el sistema local, ping utiliza el ICMP para intentar alcanzar el sistema remoto, el cual no necesariamente debe estar en la misma red. Esto significa que usted puede utilizar ping para revisar que el enrutamiento entre las redes está trabajando correctamente y para revisar si un software TCP/IP está funcionando en el sistema remoto.

netstat

Se utiliza este comando para monitorear la actividades de red en el sistema local. Usted puede monitorear diversos aspectos de la red, incluyendo el estado de los puertos locales TCP y UDP, tablas de enrutamiento IP, memoria y el uso de otros recursos del sistema y estadísticas discriminada por protocolo del sistema local.

Otras utilidades están disponibles para proporcionar formas variadas de estadística referente al software de red. A medida que las redes llegan a ser más grandes y a estar distribuida sobre areas geográficas más amplia, se ha convertido más difícil utilizar estos métodos para manejar la red. Para poder administrar las redes complejas que son hoy común, se requiere un sistema más sofisticado de adminitración de la red.

Adminitración Moderna de la Red

Las herramientas modernas de la adminitración de la red han cambiado y se han adaptado con su misión así como escalabilidad. Utilizan utilidades de administración gráficas para controlar remotamente nodos de red. La admiistración global del sistema se ha convertido más genérica, es decir, no específico a un conjunto de sistemas o a un fabricante. Cada día se requiere menos y menos conocimiento de los detalles del productos de un fabricante pero si se requiere más conocimiento en los aspectos comunes entre los sistemas. Las herramientas basadas en SNMP aprovechan estas concordancias y las realzan.

Los requisitos de una administración moderna de una red incluye los isguientes puntos:

- Administración Remota.
- Disparar alarma cuando los componentes fallan o cuando ocurren acontecimientos anormales.
- Interfaz de administración común.
- Herramientas de administración integradas.

De éstos y de otros requisitos ha surgido un sistema de protocolos de administración de red que le permiten a administradores de red el control y el monitoreo del comportamiento de la red desde una consola de

administración. El protocolo de mayor intrés es el SNMP ya que este es parte del suite de protocolos del Internet.

Componentes del SNMP

Las herramientas de administración de red basadas en SNMP tienen muchas cosas en común, incluyendo:

- Utilidad de Administración.
- Software en el nodo para hablar con el manejador (agente)
- Descripción de las características del artículo que se manejará (MIB)

Estos componentes son apoyados por tres especificaciones:

- SMI (Structure of Management Information), que describe cómo identificar y describir objetos
- MIB (El Management Information Base), cual es la información que describe los objetos, por lo general almacenada en el nodo en cuestión.
- SNMP (Simple Network Management Protocol), que define la comunicación entre los administradores y los agentes.

Entre los componentes de una herramienta basada en SNMP, se incluyen:

Managed Objects Un objeto manejado es algún componente de la red. Es un concepto general pues un objeto manejado puede ser un sistema completo, una parte de un sistema tal como un módulo de enrutamiento o un NIC. Objetos Manejados pueden ser aún más abstracto, por ejemplo una máquina de venta por moneda; de hecho, cualquier cosa que se puede agregar a una red puede convertirse en un objeto manejado.

Manager El Encargado es el software que sirve de interfaz al encargado de red humano.

Agent Un agente es una pieza de software que puede comunicarse con un objeto manejado. Responde a las peticiones del encargado para retornarle la información sobre un objeto manejado o para alterar cierto atributo del objeto manejado. Este puede también transmitir la información al encargado sin ser preguntado, por ejemplo, para indicar una condición excepcional con respecto al objeto manejado.

Para que estos componentes trabajen unidos, el sistema de administración de la red debe definir:

- Un medio de identificar los objetos manejados en la red.
- Un medio de especificar los objetos reales y sus atributos.
- Un medio de comunicarse entre el encargado y los agentes que controla los objetos manejados.

Encargado, Agentes y Nodos.

En la siguiente ilustración mostramos una posible distribución para ejecutar un sistema de administración de la red tal como la descrita. Cada uno de los sistemas en la red es un nodo manejado, con uno reservado como la estación de administración de la red. El software encargado (administrador) de red se ejecuta en este sistema.

IMAGEN PAG 378

Los nodos que son manejados no necesitan estar en la misma red que la del encargado; de hecho, el enrutador o el bridge que se utiliza para cruzarse entre las redes puede también ser un objeto manejado.

El software encargado de red puede ser basada en texto o CLI pero, pero hoy en día más comúnmente, se basa en entornos gráficos de alta calidad profesional.

SMI (Structure of Management Information)

Las especificaciones de nombre del objeto SMI se establecen en los RFCs 1155 y 1212. Los objetos son nombrados usando una estructura jerárquica. La puesta en práctica del SMI es el Management

Information Base (MIB).

Los MIBs de alto nivel son administrado por una autoridad central y la administración es delega en niveles más bajos. Los objetos en cada nivel son asignados números enteros. La identificación completa del objeto es una secuencia de números enteros. A los objetos también se le puede asignar nombres del tipo cadena, lo cual facilita la interacción con humanos.

Los objetos se nombran según el RFC 1155, “Structure of Management Information”. Esto define un esquema jerárquico para nombrar objetos, parecido al del esquema jerárquico usado en el DNS.

En cada nivel de la jerarquía, los objetos se asignan números enteros. Para especificar exactamente un objeto, usted concatena los números de los objetos que especifican una ruta con la jerarquía desde la raíz de la estructura al objeto sí mismo.

Esto puede conducir a nombres ciertamente complejos. Por ejemplo: ***1.3.6.1.2.1.5*** el cual es nombre del objeto que contiene la información sobre el ICMP.

Por esta razón, a los objetos también se les asignan nombres valorados del tipo cadena para que sean más fáciles de entender y recordar para los humanos.

La administración del espacio de nombre jerarquico puede ser dividido entre las autoridades centrales para los niveles más altos y las autoridades locales para los niveles más bajos. Esto es otra vez similar a la manera que se organiza el espacio de nombre del DNS.

Espacio de Nombre del Objeto MIB (Management Information Base)

El espacio de nombre jerárquico completo es grande. En esta ilustración que sigue se muestra la parte que conduce desde la raíz a la porción del MIB del Internet.

FOTOOOOOOOOOOOOOOOOOFOTOOOOOOOOO-380

La raíz no tiene nombre, pero si tiene tres descendientes inmediatos que representan las organizaciones de control principal.

En cada nivel, los objetos individuales se asignan números enteros. Los objetos internos en el árbol del espacio de nombre se utilizan solamente para los propósitos estructurales. Los (leaf nodes) nodos hoja contienen los atributos.

La información relacionada con Internet se puede encontrar bajo nodo que tiene el nombre de 1.3.6.1. También puede ser referida usando el nombre Internet. Los subobjectos pueden ser nombrados relativoa al objeto Internet. Por ejemplo: ***Internet 2*** se refiere al subobjecto manejado (nombre completo 1.3.6.1.2).

Internet MIB

Management Information Base, es una base de datos donde se guarda toda la información relativa a la gestión de la red. El MIB tiene una estuctura en árbol, donde en la parte superior se encuentra la información más general sobre la red, y conforma avanzamos por las ramas se consigue información más específica y detallada. Cada nodo del árbol MIB se conoce como variable, y la parte superior del árbol MIB se denomina "Internet". Aunque la ISO ha definido un MIB modelo, cada fabricante de equipos tiene el suyo propio, e el que en general aunque las variables se denominen de diferente manera la información contenida en ellas es la misma.

El RFC 1155 define la estructura del MIB principal del Internet, el sistema de los objetos que se pueden interrogar con respecto a los protocolos de TCP/IP. Éstos están situados bajo objeto del MIB nombrado 1.3.6.1.2.1. El MIB define los objetos siguientes:

System	Identificación e información sobre sistemas en la red
interfaces	Información sobre interfaces de la red
at	Conversión de dirección
ip	Información sobre IP
icmp	Información sobre ICMP
tcp	Información sobre TCP
udp	Información sobre UDP
egp	Información sobre EGP (protocolo dinámico de administración de la tabla de enrutamiento)

El MIB del Internet es administrado por el Internet Architecture Board (IAB). Todos los manejadores de SNMP entienden el MIB del Internet; por lo tanto, pueden recopilar información de los nodos manejados en la red según este modelo.

La pieza system del MIB incluye objetos como son:
system.sysdescr.0 1.3.6.1.2.1.1.1.0 system.syscontact.0 .3.6.1.2,1.1.4.0
system.sysobjectid.O 1.3.6.1.2.1.1.2.0 system.sysname.0 .3.6.1.2.1.1.5.0
system, sysupti inc. 0 1.3.6.1.2.1.1.3.0 system, sys! ocati on. 0 .3.6.I.Z.I.1.6.

Privado MIB

Los MIBs privado son utilizado por los fabricantes para definir los objetos que representan sus propios productos. Bajo el objeto llamado privado (dirección 1.3.6.1.4), el manejo del MIB se delega a los fabricantes de hardware y software de red. Los fabricantes pueden definir objetos sus el propios basados en sus propios productos.

Cuando uno de estos productos se adquiere y este debe tener soporte de SNMP (o otra sistema de administración de red), entonces este incluirá un agente y una especificación del objecto(s) MIB que representa el producto. Esta información debe ser proveída de tal manera que pueda ser alimentada al software manejador de modo tal que el manejador pueda administrar el nuevo producto.

SNMP

El SNMP es un protocolo para la comunicación entre un manejador de red y los agentes que controlan objetos manejados. Está diseñado para ser simple, introduce el menor monto de sobre peso posible sobre la red. El manejador utiliza el puerto 161 y el agente utiliza el puerto 162. El SNMP es un protocolo simple, que usa UDP como su protocolo de transporte. Todos los mensajes SNMP están diseñados para ser transportados dentro un solo datagrama UDP.

Un manejador puede solicitar información de un agente acerca de un objeto manejado o puede establecer ciertos atributos de los objetos manejados. El agente responde a estas peticiones con la información o los mensajes de diagnóstico. El agente puede también enviar mensajes no solicitados (llamados traps) para indicar condiciones excepcionales en el objeto manejado.

Mensajes del SNMP

Cinco tipos de mensajes son definidos en SNMP que permiten a un administrador interrogar o establecer el valor de una variable, permitir a un agente contestar con el valor de una variable solicitada o permitir que un agente indique cierto evento relacionado a su objeto manejado. Estos son:

GetRequest	Trae uno o más valores desde una variable específica de un objeto.
GetNextRequest	Trae el valor(es) sin especificar el nombre exacto; usando este puede iterar a través del MIB completo
SetRequest	Establece a variables específicas a valores especificados
GetResponse	Respuestas a un get o set request (simplemente llamados Response en SNMP 2 y el 3)
Trap	Indica cierto evento en el agente; enviado al manejador(es) especificado

El mensaje GetNext permite a un manejador iterar por los valores de variables en un objeto sin saber el formato exacto del objeto. Este puede ser útil para interactuar con objetos privados que no esten completamente especificados en el MIB.

Las peticiones (Request) que especifican variables utilizan el esquema de nombramiento numérico jerárquico, descrito anteriormente, para identificar las variables y los objetos. La definición de la sintaxis de los objetos describe el formato de los datos que están siendo transferidos o con el GetResponse o el SetRequest.

Los "traps" pueden ser eventos, tales como: un sistema que se inicia en frío, un interfaz de la red fallando o un problema de enrutamiento.

Comunidades

Los agentes SNMP se pueden agrupar en comunidades. Para hacer una petición (para get o set un valor) de un agente, el manejador debe saber el nombre de la comunidad y especificarlo con la petición.

Esto introduce una forma primitiva de seguridad en el SNMP. En los años más recientes, se ha efectuado actualizaciones al SNMP que permiten mejoras y refinamientos en su funcionalidad. También hay actualizaciones de definiciones para los protocolos.

El nuevo SMI, definido en el RFC 1902, amplia los tipos de datos y mejora "get" y "set". En esta manera, el SMI no fallará si uno de los valores no está disponible y el SMI puede adquirir un gran conjunto de valores a través de get-bulk.

La especificación también permite un formato mejorado de "traps" con el manejo de error mejorado.

También se provee para la administración de niveles múltiples. Esto establece una jerarquía entre los manejadores por la que un manejador general controla a los manejadores más bajos en al jerarquía usando "inform-request". Los agentes Proxy pueden también controlar redes aisladas o las redes que no son SNMP.

Lo mejor de todo es que las nuevas especificaciones son compatibles con la versión 1 del SNMP.

El SNMP 2 fué concebido para mejorar las capacidades de SNMP 1 pero aún permaneciendo compatible con las versiones anteriores.

Esta define nuevos tipos de datos, mejora "get" y establece los mensajes, proporciona un mejor manejo de errores para las "traps" y define niveles del control de la administración de los sistemas del SNMP.

Desafortunadamente, la única área en la cual el SNMP 2 no fué éxitoso consiste en completamente definir una especificación apropiada de seguridad para substituir el uso de las cadenas de la comunidad como mecanismo de la seguridad.

Versión 3 del SNMP

En esta sección destacamos algunas de las características de la versión 3 del SNMP, establecido en los RFCs 2271, 2272, 2273, 2274, y 2275. El SNMP 3 es la última versión de SNMP. Su objetivo principal es proporcionarle SNMP 2 y seguridad avanzada.

Con este fin, le agrega al SNMP 2, integridad de los datos, autentificación, cifrado (encriptación) y tecnologías claves de administración, todo esto junto con algunas mejoras administrativas.

Es muy probable que el SNMP 3 sea la última versión del SNMP por un largo tiempo. Esta disposición fué lanzada en el 1998 y ya existen muchos dispositivos que soportan SNMP 3.

La versión 3 agrega seguridad avanzada y cifrado atraves de la integridad de los datos, autentificación, ataques de enmascaramiento, así como MD5 y Secure Hash Algorithm. Agrega privacidad a traves de del soporte para DES enriptación de llaves privadas, autorización y el control de acceso.

Además, la versión 3 soporta el cifrado de llaves públicas, administración de llaves de seguridad y la interoperabilidad de múltiples versiones. Es compatible con las versiones 1 y 2.

Usar el SNMP

Hay básicamente dos tipos de utilitarios SNMP: herramientas que se ejecutan en la línea de comandos (e.j., snmpget y snmpset) y los productos con GUIs.

Los productos gráficos son casi siempre de muy alta calidad. Estas gráficas por lo geeneral se ejecutan en estaciones de trabajo. Estas herramientas con GUIs ofrecen varias características que las hacen muy útiles, ofrecen fácil manejo y aprendizaje y hacen un magnifico interface a las herramientas SNMP. Sin embargo, estos productos GUIs privados tienden a ser costosos. Algunos ejemplos incluyen NetManager del sun, a NetView/6000 y Tivoli de la IBM, y HP OpenView.

Otros Protocolos De la Dirección De la Red

El SNMP, aunque el estándar más robusto y extensamente el más usado para la solución de problemas y la representación de la red, este no es de ninguna manera la única herramienta de administración de la red disponible. Otros incluyen CMIP, CMOP y NetView.

La ISO está desarrollando un protocolo de administración de red conocido como CMIP (Common Management Information Protocolo) que es muy similar al SNMP pero que trabaja con la pila de sieta capa de la ISO. CMIP también está disponible en las redes de TCP/IP, conocido como CMOT; sin embargo, esto ahora ha sido adoptado y sobre pasado por eventos y substituida por SNMP.

Hace ya muchos años que la IBM ha tenido una estrategia de la administración de la red como parte del SNA, ésta era conocida como NetView (no debe ser confundida con los productos NetView/6000 que so basados en el SNMP). A medida que el SNMP es más aceptado, los productos NetView ahora son más integrado con el SNMP.

Observe que las definiciones de SMI y de MIB no son tales que están atadas exclusivamente al SNMP. Si se utilizan metódos de administración de redes alternativos, las específicaciones de los mecanismo de los objetos pueden ser utilizadas donquiera.

Ejercicio 5-2: Implementación del Nuevo Producto de Administración de Red.

No se proveen soluciones para este ejercicio.

Su tarea como administrador es planificar la implementación del nuevo producto en toda la sede principal de su empresa. Le será necesario tomar en consideración los siguientes puntos:

1.- ¿Cuáles parámetros del sistema y de la red dene usted medir?

2.- Cuáles son los nodos críticos en la red existente en la compañía en este momento?

3.- ¿Dónde tiene sentido colocar la estación de trabajo que efectuará la administración?

4.- ¿Existen algunas situaciones de red comunes que usted puede programar cierto tipo de respuesta automatizada?

RESUMEN

En este capítulo, examinamos los siguiente:

- Solución de Problemas
  - Métodos metódicos de la Solución de Problemas
  - Los archivos protocol(s), services y inetd.conf
  - Herramientas y métodos útiles para la solución de problemas de la red que incluyen ping, traceroute, netstat, ifconfig, arp, nslookup y hostname.
  - El ICMP y los tipos de mensaje del ICMP

Administración de la Red
  - Varios métodos y razones para la administración de la red
  - Componentes y estándares de SNMP
  - La necesidad del SNMP y de otras herramientas de la administración de la red

PREGUNTAS

Las respuestas a estas preguntas está en el Apéndice A.

1.- ¿Cómo puede usted comprobar si cualquier parte del TCP/IP ha registrado un error?

2.- ¿Cómo puede usted probar su espacio de nombre (namespace) de dominio?

3.- ¿Cuáles son algunas herramientas para la administración de una la red moderna?



DNS (Sistema de Nombre de Dominio)

OBJECTIVOS

Al finalizar este capítulo, usted podrá:

- Instalar y configurar los servicios básico de DNS.
- Entender y ser capaz de implementar un Servidor de DNS.
- Detallar las razones dettras de la implementación de la identificación de un DNS, el uso de DNS para la identificación en los procesos de busquedas y los pasos tomados en el proceso de resolución de nombres así como la búsqueda inversa.
- Definir los elementos de un DNS y sus interacciones.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. De dos razones del porqué el servicio de nombre no es efectuado en una forma centralizada.

2. ¿Cúal es el nombre del directorio raíz del árbol del DNS?

3. ¿Qué es un FQDN (Fully Qualified Domain Name)?

4. ¿Qué es un resolver?

5. ¿Son las siguientes sentencias falsas o verdaderas?
   - El DNS es independiente de la topología de la red.
   - Un resolver siempre sabe la localización del servidor de nombre ráiz/root.
   - El resolver efectúa un cache de la data.

INTRODUCCION

En este capítulo cubriremos las teorías del servicio de nombres y responderemos las inquietudes de porque usar un servicio de nombre. En principio cubrimo lo que es el DNS (Domain Name System), el cual es usado en las mayorías de los sitios web conectados al Internet; pero si consideraremos otros servicios de nombres de Internet usados muy escazamente como es el WINS. En este capítulo también describiremos lo que son dominios, namespaces y zonas de los dominios como también explicaremos como trabaja un DNS. También se discutira los tipos de servidores, record de recursos y la resolución de un nombre.

DNS A SIMPLE VISTA

El DNS es el mecanismo por el cual las direcciones IP son mapeadas a nombres de los ordenadores. Los siguientes tópicos son discutidos en esta sección:

- Servio de Nombre
- DNS
- Records de Recursoso
- Resolvers/Resolución
- Cache para mejorar Rendimiento
- Mapeado de Direcciones a Nombres
- Relación DHCP y DNS
- Actualización Dinámica
- Configurar y Ejecutar un DNS
- Otros Servicios de Nombres

Servicio de Nombre

Un servicio de Nombre es un servicio de información que mapea o convierte nombres a direcciones IP. El servicio es accesado transparentemente; al comunicarse con un host remoto, el usuario no se percata del procedimiento usado por el servicio de nombre para encontrar la dirección IP.

Un servicio de nombre traduce un simple nombre a una dirección IP.

codigolibre.org ========>>>>> 206.42.200.137

No es suficiente poderse comunicar usando direcciones IP unicamente. Aunque la comunicación con el uso de IP es totalmente posible, las direcciones IP son dificiles de recordar para las gente y tienden a cambiar, por ejemplo, si un equipo es cambiado de una red a otra aunque dentro de la misma LAN. Otro problema con el uso de direcciones IP en el futuro es con el cambio del protocolo IP al ipv6.

Los nombres nos libran de las restricciones que colocan los protocolos, y aún más importante, son más fáciles de recordar que las direcciones IP numericas.

Los Problemas de un Espacio de Nombres Plano

En los dias de un Internet con sólo unos cientos de hosts, los nombres de estos host eran administrados por una autoridad central. El administrador de sistemas contactaba el Centro de Información de Redes con siglas de NIC (Network Information Center) con el nombre del nuevo host. El NIC procedería a efectuar una búsqueda en su base de datos de nombres en el archivo HOSTS.TXT, para determinar si el nombre ya existe. Si el nombre es único y no existe entonces se le asigna ese nombre al nuevo host y se añade al archivo HOSTS.TXT seguido de la dirección IP.

El archivo HOSTS.TXT luego era propagado a todos los sitios de Internet usando el servicio de ftp. Los sitios se conectaban regularmente para proceder a descargar una copia fresca con los ultimos cambios.

En resumen, los problemas con este mecanismos son:

- **Existe una gran posibilidad de conflicto de nombres.**
  Elejir un nombre único es difícil las mayorías de las veces. Los nombres son limitados y tenemos que ser muy creativos para poder seguir supliendo nombres únicos. Este problema se agraba a medida que se incrementa el número de hosts en el Internet.

- **La administración centralizada de los nombres se convierte en un problema.**
  Este no es sólo un problema de la autoridad administradora sino de comunidad en general. Nadie desea tener que esperar para que una nueva elección de un nombre sea declarada única y luego tener que descargar un nuevo archivo hosts. El el caso de la NIC no existía tal administración de los nombres.

- **Los problemas de distribución se incementan dramáticamente.**
  Con todo el mundo teniendo que descargar el archivo hosts por lo menos todos los dias, se producía una carga innecesaria sobre la red y el archivo hosts se convirtió intolerable. El mantenimiento de su consistencia se torno imposible. Cuando descargabas el archivo HOSTS.TXT, ya otro host había sido agregado.

DNS

El DNS es una base de datos que mapea nombres a direcciones. Esta base de datos es distribuida en todo el Internet, independiente de la topología de la red. Aquí podemos ilustrar la traducción de un nombre de un ftp a una dirección IP.

ftp.codigolibre.org ================> 200.42.200.17

La función principal del DNS es la distribución y su consecuente ventajas son:

- Los hosts no tienen que descargar la dirección correspondiente a cada host en el Internet; ellas solo deberan ubicar la dirección de la máquina con la cual desean comunicarse en el momento necesario.
- La distribución de la administración: si uste agrega un nuevo ordenador a su red local o si sólo le cambia su nombre, usted no tendrá que informarle a la agencia central. Simplemente tendrá que alterar la base de datos de us DNS.
- Conveniencia.- Nombres conocidos por el usuario son más fácil de recordar que sus respectivas direcciones IP.
- Consistencia.- Las direcciones IP pueden cambiar pero los nombres permanecen constantes.
- Simplicidad.- Usuarios necesitan aprender solo un nombre para encontrar recursos ya sea en Internet o en una Intranet.

Espacio de Nombre de Dominio del DNS

Provee la estructura jerárquica de una Base de Datos distribuida. Cada dominio tiene un único nombre. Los hosts de Internet están organizados en un árbol único ya que la jerarquía es independiente de la topología de la red fisica. El root domain esta en lo alto y se representa por un punto. Bajo el root domain se encuentra el top level o first level, puede ser un tipo organizacional como .org, ó una localización geográfica como .do. El Second Level son registrados para organizaciones individuales tal como kernel.org. En el second level pueden contener muchos subdominios y algún dominio contiene hosts (host que viene a ser una computadora específica dentro del dominio). Un FQDN (Fully Qualified Domain Name) describe la relación exacta de un host a su dominio.

El término name space (o espacio de nombres) se refiere a todos los nombres posibles. Un namespace plano limita este conjunto, mientras que uno jerarquico permite un conjunto a escojer de nombres casi ilimitados. Al trabajar con los DNS concentrese en los namespace y no las redes subyacentes.

Una Jerarquía de Dominios

La mejor manera de describir un dominio es como un sub-árbol de un namespace. Cada nodo en el subárbol es nombrado por el una etiqueta.

Existen dos convenciones usadas para elejir dominios, organizacional y geografica:

Organizational:

com	Organizaciones Comerciales (como es http://www.ibm.com)
edu	Organizaciones Educativass (http://www.see.edu)
gov	Organizaciones Gubernamentales (http://www.nasa.gov)
mil	Organizaciones militares (http://www.army.mil, http://www.navy.mil)
net	Organizaciones de Redes (http://www.verizon.net)
org	Organizaciones No-Comerciales (http://www.codigolibre.org)
int	Organizaciones Internacionales (http://www.nato.int)

Geographical:

do	Republica Dominicana
mx	México

Se permite usar mayusculas y minusculas para registrar l;os dominios pero esto no significan nada ya que todas son tomadas como minusculas, lo que significa que se escribes dos nombres uno toda minusculas y otro con mayusculas y minusculas se trataran como el mismo nombre. La etiqueta debe empezar con una letra y puede terminar con una letra o un digito, y sus caracteres de interior deben ser todos letras y digitos o guiones. La etiqueta no debe ser más de 63 caracteres.

Nombres de Dominio

Un nombre de dominio esta compuestop de todas las etiquetas desde la raíz, escrito desde la derecha hacia la izquierda, separado por un punto (.). Un nombre que termina con la raíz (root) es denominado como un FQDN (Fully Qualified Domain Name). Una etiqueta debe ser único dentro de su dominio padre.

Los nombres de dominios pueden ser especificados de dos maneras:

- Los nombres absolutos son expresados en relación a la raíz, asi que terminan con la etiqueta de la raíz, un ejemplo es: ftp.linux.org.
- Los nombres relativos representan la última etiqueta de un nombre de dominio incompleto; ellos son relativos a un origen bien conocido (normalmente es la raíz del dominio) o a una lista de dominios usada como una lista de búsqueda.

Fijese como la raíz del dominio está a la derecha. Los nombres de dominio se escriben de atras para alante, desde el más específico (el nodo más bajo) hacia el menos específico (la raíz).

Servidor de Nombres

Los servidores de nombres son los repositorios de información que hacen la base de datos. Cada nodo en el árbol tiene cierta data asociado con el. Un servidor de nombre (en inglés, nameserver) almacena la información para el espacio de nombre, el namespace. Se dice que es autoritario para la parte del namespace de la cual el administra y contiene toda su información.

La tarea esencial de un servidor de nombre es responder las peticiones basado en la información que el contiene.

Delegación

Cuando la administración de dominio.org ha sido delegada a Compañía X,entonces dominio.com puede delegar la administración aún más.

Al delegar administración, estamos realmente delegando lo siguiente:

- El poder de decisión sobre la elección de nombres; la elección de los nombres de subdominios no esta sujeta a elegirlos únicos globalmente; los nombres deben ser únicos sólo dentro del subdominio al cual pertenecen.
- El almacenado y mantenimiento de los nombres dentro de esa parte del espacio de nombre del dominio (namespace).

Zonas/Zones

La información accesada via el espaciode nombre de dominio es rganizada en unidades llamadas zonas/zones. Una zona contiene toda la información acerca del dominio excepto la información delegada a otros servidores de nombres. Un servidor de nombres puede ser la autoridad de muchas zonas. Aún más, estas zonas no necesariamente deben encontrar adjecentes dentro del espacio de nombre de dominio.

El servidor raíz, por ejemplo, sirve subdominios como son .org asi como sirve el dominio raíz (pero, si observan los dominios de los países como el .do); los mismos principios son usados para mapear direcciones a nombres.

El uso de los términos DNS se torna un poso confuso; las personas a menudo se refieren a las zonas como dominios.

Es requerido que los dominios accesibles desde el Internet sean soportados por lo menos por dos servidores de nombres. Estos es sólo por razones de confiabilidad. A menudo, los administradores de sistemas han obviado esta regla y han sido blanco de castigo por la comunidad que rigen el Internet.

Servidores Primarios y Secundarios

La información de una zona es replicada atraves de por lo menos dos servidores para garantizar confiabilidad. La data de la zona debe estar disponible aún cuando un servidor fracaza o durante un ruptura en la red. Los servidores primarias son a veces referidos como los servidores maestro/master.

Toda la data origina en los archivos maestro o los archivos de data que se encuentran repartidos en todos los hosts que usan el sistema de dominio. Estos archivos maestros son actualizados por los administradores del sistema local.

Los cambios de una zona son efectuados en el servidor primario. El servidor carga la copia maestra de su data desde los archivos maestro cuando este se inicia.

Un servidor secundario mantiene una copia de la data de la zona. Este revisa periódicamente el servidor primario para asegurarse que sus zonas están actualizadas. Si no lo están, este adquiere una nueva copia desde el servidor primario de las zonas actualizadas. En realidad los servidores no son de un tipo o de otro; un servidor puede ser el master de una zona y el secundario de otra.

Un cliente que peticiona información desde un servidor de nombres no esta conciente si ese servidor de nombres es un primario i secundario en esa zona.

Otros Tipos de Servidores

Hay tres tipos de servidores: solo capturadores/caching, reenviadores/forwarders y esclavos/slaves. Todos los servidores hacen cache de data, pero un servidor caching-only no es quien autoriza para esa zona. Este toma la carga de otros servidores atraves de la construcción de un cache para poder responder las peticiones. Un reenviador/forwarder maneja las peticiones en nombre de un servidor de nombre cliente. No se necesita ninguna configuración especial; todos los servidores son forwarder por naturaleza ya que otro

servidor les envia peticiones recursivamente. Esto es útil cuando las conexiones a redes externas son lentas o quizas costosas. Los servidores esclavos son muy parecidos a los forwarder excepto que no sale fuera de la red local si no hay una respuesta desde el forwarder. Esto es muy útil cuando no hay conexión a la red externa o cuando la máquina pasarela/gateway esta actuando como un firewall.

Records de Recursos

Toda la información almacenada o transmitida dentro de un DNS se encuentra en un formato estandar llamado Resource Record (RK). El RK fundamentalmente esta dividido en cuatro parte:

Domain	El nombre de un dominio al cual el record se refiere
Class	La clase del record (por lo general es IN para el Internet)
Type	El tipo de record, e.j., para que es la información
Information	La data para este record

En este ejemplo ilustramos el significado de cada parte de un record de recurso o RK.

200.42.212.137	**-ADDR.ARPA.**	**IN**	**PTR**	**SRI**	**-NIC.ARPA.**
domain		**class**	**type**		**information**

Tipos de Records de Recursos

Los DNS definen varios tipos de records de recursos. Entre los más comunes se incluyen:

- **A (dirección IPv4)**
  Indica la dirección que corresponde a un host en particular
- **AAAA (dirección IPv6)**
  Indica la dirección que corresponde a un host en particular
- **NS (Nameservcr)**
  Indica un servidor de nombre responsable de un dominio en particular
- **SOA (Start of Authority/Inicio de la Autoridad)**
  Designa el inicio de una zona
- **PTR (Puntero/Pointer)**
  - Permite nombres especiales a que apunten a otras localidades en el dominio
  - Usado principalmente para el mapeado de direcciones a nombres

Esta lista en ninguna forma es completa y existe muchos más tipos de records, aqui sólo mencionamos unos cuantos para ilustrar.

Primero se necesitan records de Direcciones/Address, los cuales mapean nombres a direcciones- la verdadera función de un DNS. De segundo, necesitamos records de Servidor de Nombres, con los cuales un servidor de nombres para el dominio en cuestión. Tercero, necesitamos un record de Inicio de Autoridad (Start of Authority) el cual designa el comienzo de una zona; este también contiene la información relacionada con la propagación de la información de la zona entre los servidores primarios y secundarios. Y para finalizar, los records de Punteros/Pointer serán usados para mapear las direcciones a nombres.

Resolvers

El resolver es como el cliente procesa entrada en la base de datos del DNS. Este extrae la información desde el servidor de nombres en respuesta a las peticiones del cliente. Para lograr esto, este debe saber como llegar a por lo menos un servidor de nombres.

El resolver normalmente es implementado como un conjunto de librerias vinculadas con aplicaciones que necesitan comunicarse usando nombres de equipos. Una aplicación simplemente llama una rutina que resuelve/resolver y espera que esta retorne: esta no se alerta de la existencia de la peticiones del servidor nombres que el resolver esta iniciando para su beneficio.

Resolución de un Nombre

Un proceso cliente desea comunicarse con otro proceso en ftp.descarga.abiertos.org. Para poder efectuarlo, se requiere la dirección IP de esa máquina en particular, y no solamente el nombre. Si se usa las API de sockets, nuestro proceso cliente llama la función gethostbyname(), la cual es parte de la libreria resolver.

1. La función resolver envia una petición a servidor de nombre local (el resolver determina a cual servidor de nombres usar buscando en su archivo de configuración local). La petición específica el dominio de interes (ftp.descarga.abiertos.org) y la información requerida (un record del tipo Address).

2. El servidor de nombre local primero revisa su propia información para ver si poseé la respuesta a la petición. El caso negativo, este envia una petición por el dominio ftp.descarga.abiertos.org al servidor raíz/root. En el caso que el servidor raíz tampoco tenga la información, si sabe de un servidor más cercano al domio de interes, asi pues responde con una referencia/referral a un servidor del dominio .org.

3. El servidor de nombre local envia una petición por el dominio ftp.descarga.abiertos.org al servidor de .org. Este servidor no tiene una respuesta y responde con una referencia/referral a un servidor en el dominio abiertos.org.

4. El servidor local envia una petición por el dominio ftp.descarga.abiertos.org por abiertos.org. Este servidor no tiene una respuesta y responde con una referencia a un servidor para el dominio descarga.abiertos.org.

5. El servidor local envia una petición por el dominio ftp.descarga.abiertos.org por abiertos.org. Este servidor si contiene la información requerida (lo que en este caso signfica que tiene un record del tipo Address para ftp.descarga.abiertos.org). El servidor retorna la dirección al servidor que hizo la petición.

6. El servidor de nombre local responde a la petción original del resolver.

El rutina resolver gethostbyname() retorna al proceso cliente la dirrección de ftp.descarga.abiertos.org.

Uso de Cache para Mejorar Rendimiento

Al igual que en un procesador o un disco duro, el uso de cache en un DNS mejora el rendimiento. El rendimiento mejora dramáticamente ya que el cache es revisado primero para responder una petición. El problema que se presenta con el cache, claro como con todo las tecnicas de cache, es que si la data original del servidor de nombre ha sido actualizada puede existir un retardo ya que la data en cache es una copia local. Este servidor responderá peticiones que puede estar desactualizada. Esta data es conocida como no autorizada. Los servidores primarios y secundarios son la fuente original de la zona y retornan data que si es autorizada. Un servidor secundario está autorizado para la zona, ya que este contiene la información total de esta zona.

Existe un mecanismo de timeout/vencimiento para eventualmente descarta data en cache. Asociado con cada record de recurso llamado TTL (Time to Live, tiempo de vida) que indica que tiempo le resta a esta data en cache antes de ser descartada.

Los diseñadores del DNS consideran que el acceso a la información es más crítico que actualizaciones instantáneas o garantía de consistencia. La mayoría de la data en el sistema cambiará lentamente, pero el sistema debe poder manejar los subconjuntos que cambian más rápido, en el orden de segundos y minutos.

Mapear Direcciones IP a Nombres

Algunas aplicaciones necesitan saber el nombre del ordenador al ser entregado la dirección IP. Este mapeado es efectuado usando el dominio especial, IN-ADDR.ARPA. Este dominio es configurado especialmente para mapear direcciones a nombres. Las direcciones en este dominio corresponden a direcciones IP pero aparecen alreves, porque esta es la manera en que los nombres de dominios de los DNS se escriben. Los nodos en este dominio son nombrados por los bytes de su dirección IP. Colocar imagen del árbol de nombre y su relación con IN-ADDR.ARPA.

Las aplicaciones y servicios que se ejecutan hoy requieren que un host que se conecte a ellos pase busquedas en reverso (reverse lookup), un término usado para describir el proceso de mapear direcciones IP a un nombre, para asi poder establecer una conexión. Los hosts que no pasan la búsqueda inversa pueden que tengan un error lijero de configuración o pueden ser mal configurados a propósito para enmascarar la identidad de alguien intentando violar nuestra seguridad o ejecutar un ataque DoS (Denial of Service).

Relación DHCP y DNS

Las siglas DHCP significan Dynamic Host Configuration Protocol, que en castellano es Protocolo de Configuración de Hosts Dinámica. El propósito del DHCP es permitir que un ordenador en una red IP extraigan su configuración desde un servidor o servidores, en especial servidores que no tienen información exacta del ordenador en particular hasta que la petición sea efectuada. El propósito general del DHCP es reducir el trabajo necesario para administrar una red IP grande. El servidor DHCP puede ser configurado entregar un número de IP al azar o un nombre de dominio a equipo cliente por un tiempo en específico, denominado a un alquiler. Una vez se apaga el cliente y el periodo de tiempo de alquiler expira, el servidor DHCP puede re-alquilar a otro cliente el mismo IP. Una excepción a esta regla es cuando un IP estaticco es asignado a un cliente en particular. Si el servidor DHCP asigna un nombre de dominio a un computador cliente, se necesitará un servidor DNS para resolver la dirección IP apropiada.

Actualización Dinámica

El término actualización dinámica es usado para describir el proceso de actualizar automáticamente los records. El RFC 2136 detalla el procedimiento dinámico de actualización. Si una zona está configurada para utilizar actualizaciones dinámica, es entonces importante no editar el archivo de zona manualmente. Si se presenta esta necesidad, es de suma importancia entonces, apagar el servidor de nombre y editar el archivo de zona y eliminar el archivo de diario (journal) de la zona antes de reiniciar el servidor de nombre.

Los pasos del maestro primario para autenticar la petición a quien peticiona son:

1. Un mensaje de petición actualizada es enviada desde un cleinte a un servidor local.

2. El mensaje actualizado será reenviado al Servidor Maestro Primario.

3. El Maestro Primario revisa los prerequisitos especificados en el mensaje.

4. El equipo que efectua la petición es validado (dependiente de la implementación hasta que las extensiones de seguridad son incluídas, probablemente en la dirección de quien efectua la petición). Luego de actualizar su base de dato, el servidor la escribe a memoria no volátil (o resescribiendo el archivo maestro original o en un diario/log actualizado). El SOA serial es incrementado.

5. Si se implementa el RFC 1996, el servidor puede enviar mensajes “DNS NOTIFY” a los servidores esclavos.

Configurar y Ejecutar un DNS

Decidir ejecutar un DNS depende en un número de factores. Si usted está conectado al Internet global, deberá ejecutarlo. Aunque es obligatorio ejecutar el protocolo para usar el Internet, sin el DNS fuese imposible usar los recursos del Internet sin el. Si esta planificando conectarse al Internet en el futuro es recomendado registrar su direcciones de red y empezar a ejecutar el DNS.

Si usted no esta conectado al Internet, puede que aún asi decida ejecutar DNS si su red interna (internetwork) es lo suficientemente amplia, dividida sobre una área geográfica amplia o redes por separado, o si es administrada por diferente grupo de personas o divisiones lógicas.

Para ejecutar un DNS cuando usted no está conectado al Internet, deberá configurar su propio servidor de nombre raíz/root. Estos serán contactados como parte del proceso de resolución.

Otros Servicios de Nombre

Hay distintos métodos disponibles para la resolución de nombres parecidas al de los DNS. WINS es uno de estos métodos alternativos de resolución de nombre que esta basado en NT pero también disponible en GNU/Linux como característica del suite de Samba. El protocolo LDAP (Lightweight Directory Access Protocol) es para proveerr un servicio similar al servicio de directorio NDS (NetWare Directory Services) o el Active Directory. Aunque estos servicios de directorios son todos interoperativos con el estándar X.500 para los servicios de directorios, pero NetWare y WINS incluyen características de componentes propietarias, sin embargo existe una implementación completamente libre y Open Source de LDAP, de nombre OpenLDAP, disponible para GNU/Linux.

WINS

Puede que encuentre en redes ejecutando servidores sobre plataforma PC, el servicio WINS. Directamente del nombre se puede deducir la empresa de la cual se origina y sobre que clientes y servidores se ejecuta casi en exclusividad. Los servidores WINS, por lo general equipos NT, resuelven peticiones de direcciones IP en la misma forma que los servidores DNS. Fué creado para un uso primordialmente dentro de redes LANs y con el crecimiento de redes distribuidas hoy dia es muy poco usado.

WINS tiene muchas similitudes con el DNS en la menera que este comparte información de manera regular y en la menra que la información caduca si no es refrescada. Fué diseñado para ambiente dinámicos y en este respecto comparte sus características con el DHCP (Dynamic Host Configuration Protocol).

Un servidor WINS aceptará un registro dinámico de un cliente y procederá luego a propagar el nombre a los servidores locales, si el cliente no re-establece registro de su nombre por un período de tiempo establecido, será removido del espacio de nombre del dominio. Con el uso en combinación de DHCP y WINS, un cliente puede ser removido desde una red y reconectado a otrra sin la necesidad de ser reconfigurado manualmente.

ACAP y LDAP

La IETF (Internet Engineering Task Force) dirige el desarrollo de nuevos protocolos para proveer una mejor estructura organizativa del Internet y sus usuarios. Uno de estos protocolos es el ACAP (Application Configuration Access Protocol). Este le concede acceso a las aplicaciones como los clientes de correos a otros servicios como digamos los libros de directorios de direcciones.

En este momento, el Internet no poseé un servicio de requisición a directorio para que el ACAP acceda. El DNS proveé el servicio de una búsqueda por el nombre de un host y nada más. La especificación OSI X.500 es un servicio completo de servicio de directorio que incluye un modelo de información, espacio de nombre y una infraestructura de autentificación pero si se le agrega al Directorio Access Protocol es muy

pesada para su implementación global.

El LDAP (Lightweight Directory Access Protocol) simplifica muchas de las operaciones X.500 y puede ejecutarse sobre las redes existentes TCP/IP. Esto hace que el X.500 este disponible sobre una variedad más amplia y el LDAP puede hasta ser ejecutado nativamente con su propia base de datos. LDAP proveé ACAP con una estrucutura de directorio. Esta puede prover libros de direcciones, tanto para compañías privadas como libros públicos basaados en servidores como el de Netscape (servidor de directorio) y four11 con su página web en (http://www.four11.com). LDAP también puede integrar redes TCP/IP con servicios propietarios de directorio commo son el de Novel NDS y Banyan StreeTalk.

CONFIGURAR UN SERVIDOR DNS

En esta sección nos encaminaremos por los pasos necesarios básicos de configurar los aspectos variados de un servidor DNS. Cubriremos los aspectos prácticos de la implementación de un servidor DNS. Discutiremos los parámetros pertinentes a su dominio. También cubriremos la escritura a los records de recursos, instalar y configurar el software del servidor y depurar el servidor de nombre.

Los siguientes tópicos serán discutidos en esta sección:
- Instalar y Configurar un Servidor de Nombre DNS
- El BIND (Berkeley Internet Name Daemon)
- El Archivo de Inicio del Servidor de Nombre- BIND 4
- El Archivo de Configuración del Servidor de Nombre -BIND 8
- Archivos de Data (Archivos Maestros)
- Configurar el Servidor de Nombre
- Información para Depurar

Instalar y Configurar un Servidor de Nombre DNS

Existen tres componentes necesarios para configurar y ejecutar un servidor de nombre. Estos son el software del servidor de nombre, el archivo de arranque/boot y los archivos maestros (archivos de data). El software del servidor de nombre de GNU/Linux es llamada named. El archivo de boot puede que no sea necesario en todos los sistemas. El archivo maestro contiene toda la información del dominio, el mapeado inverso de las direcciones para las direcciones IP del dominio, el mapeado inverso de las direcciones para el loopback y el archivo cache.

El servidor de nombre DNS es implementado ejecutando el daemon de nombre named, también conocido como el BIND.

El Daemon de Nombre de Internet de Berkeley (BIND)

Desarrollado en la Unviversidad de California en Berkeley, el BIND se ha convertido en el servidor de nombre más usado. El es proveído en muchos ambientes y es parte integral de las mayoría de versiones o distros de GNU/Linux.

El BIND se encuentra ya bajo el control de la ISC y es mantenido y desarrollado activamente. La ISC también tiene otros proyectos, como un servidor de DHCP este puede ser encontrado en la página web http://www.kc.org/. La versión 4 del BIND no está siendo desarrollada ni mantenida actualmente. La revisión actual de la versión del BIND 4.9.8, pero esta es sólo una parte de la serie de correción de fallas/bug-fix.

Los desarrolladores y mentenedores de BIND, la ISC, recomiendan en lo absoluto que los usuarios migren hacia en BIND 8.x, el cual tiene muchas mejoras y características sobre la serie antescedente 4.x, y mucho menos problemas de seguridad y ofrece DDNS (Dynamic DNS). La versión 8 de BIND, DDNS, fué lanzada en mayo 1997; esta implementa actualizaciones dinamicas, notificación de cambio de DNS y

ambientes dinámicos. Esto significa que los servidores maestros pueden ahora notificar a los esclavos de cambios ocurridos. Además, las extensiones de sseguridad de DNS usan firmas digitales para autentificación y aplicar estas actualizaciones de seguridad.

El soporte de IPv6 fué introducido por primera vez en la versión del BIND 4.9.4 y se continúa mejorando en el árbol de la versión 8. El árbol más reciente es el del BIND 9 que fué lanzado estable y se encuentra en la versión del BIND 9.1.0. El BIND es configurado usando un archivo de arranque/boot y archivos de data local. El archivo boot define:

- El directorio en el cual se ejecutará el servidor de nombre
- Las zonas en las cuales el servidor de nombres es primario o secundario y el archivo que continen la data del servidor de nombre.

Los archivos de data contienen los records de los recursos para todas las zonas que el servidor de nombres es autorizado.

Archivo Boot del Servidor de Nombre- BIND 4

Aquí le presentamos un ejemplo de un archivo boot para la zona abiertos.org del servidor primario de un BIND 4.x.x:

```
;
; Archivo /etc/named.boot para abiertos.org
;
directory               /var/named
;
;                       dominio             archivo          o host
;----------------------------------------------------------------------------------------------------------
cache                   •                   root.cache
primary                 abiertos.org        db.abiertos.org
secondary   descarga.abiertos.org     200.42.200.47    descarga.backup
```

Hay dos versiones diferentes de BIND comúnmente usada al día de hoy, aunque ya esta disponible la versión 9, que son: la serie BIND 4.x.x, la cual ha está en uso por mucho tiempo y la más reciente producida por la ISC, la serie BIND 8.x.x. La serie 4.x.x ya no está en desarrollo y se le aconseja a los usuarios migrar a la serie BIND 8.x.x. Esta nueva versión es mucho más configurable y tiene muchas mejoras de seguridad y correcciones de falla. Si se encuentra en la necesidad de convertir un archivo named.boot al nuevo formato, puede usar el utilitario de nombre named-bootconf que se encuentra dentro del paquete de BIND 8.

Debido a que el formato del archivo de control del BIND cambia entre las versiones 4.x.x y 8.x.x, le mostramos ambas variantes aquí.

Este archivo de arranque indica que:

- El aservidor encontrará su archivo de configuración en el directorio /var/named .
- El servidor cargará con la data del dominio root/raíz, el cual se almacena en el archivo root.cache.
- Este es el servidor primario para la zona abiertos.org, y el recurso de record para esa zona deben ser cargados desde el archivo db.abiertos.org.
- Este servidor también es un servidor secundario para descarga.abiertos.org (el cual se presume tiene su propio servidor primario); la data de la zona es cargada atraves de una transferencia desde el host 200.42.200.47 al momento de iniciar el servidor, y se mantiene un backup en el archivo local descarga.backup.

Archivo de Configuración del Servidor de Nombre- BIND 8

Aquí le presentamos un ejemplo de un archivo de configuración para la zona abiertos.org del servidor primario de un BIND 8.x.x:

```
;
; Fichero de zona para abiertos.org
;
; Minimo indspensable para tener funcionando un dominio
;
@   IN          SOA   sn.abiertos.org.            hostmaster.linux.bogus.   (
                      199511301                   ; Número de serie, fecha de hoy+n de serie de hoy
                      28800                       ; Tasa de Refresco, en segundos
                      7200                        ; Tasa de Reintento, en segundos
                      3600000                          ; Caducidad para secundario, en segundos
                      86400 )                     ; Tiempo de Validez para Clientes, en segundos
                NS    sn.abiertos.org.
                NS    sn.backup.abiertos.org.
                MX    10 mail.abiertos.org.       ; Intercambiador de Correo Primario
                MX    20 mail.backup.abiertos.org.    ; Intercambiador de Correo Secundario
localhost       A     127.0.0.1
ns              A     127.0.0.2
mail            A     127.0.0.4
```

Deben de observarse dos cosas sobre los registros SOA. sn.abiertos.org debe ser una máquina actual con un registro A. No es legal tener un registro CNAME para la máquina mencionada en el registro SOA. Su nombre no necesita ser sn, podría ser cualquier nombre legal de máquina. A continuación, en postmaster.abiertos.org deberá aparecer algo como postmaster@abiertos.org; esto sería un alias de email, o una cuenta de correo, donde la(s) persona(s) que realizan el mantenimiento de DNS deberían leer con frecuencia el correo. Cualquier email respecto del dominio ser´a mandado a la dirección aquí indicada. El nombre no tiene por que ser postmaster, puede ser cualquier dirección email legal, pero la dirección email postmaster funcionará bien.

Hay un nuevo tipo de RR en este archivo, el MX, o Mail eXchanger. Este indica el sistema de correo a donde mandar el correo dirigido a alguien@abiertos.org, pudiendo ser tambíen mail.abiertos.org o mail.backup.abiertos.org. El número que precede a cada nombre de máquina es la prioridad del RR MX. El RR con el número más bajo (10) es aquel al que el correo sería enviado primero. Si este falla, puede ser mandado a otro con un número más alto, que será gestor secundario de correo, como mail.backup.org que tiene una prioridad 20 aquí.

Con el BIND 8, el archivo de configuración es por lo general llamado named.conf y no named.boot, que se nombró en la versión 4 de BIND. Fijese también que el caracter de comentario ha cambiado del anterior (;) a nuevo más estandar antoración de comentario (#).

Como podrá notar el sintaxis del archivo de confiuración principal del BIND 8 es muy diferente. La suma de muchas otras opciones posibles entre las llaves permiten una gran variedad de posibilidades de configuración que no era posible en el pasado y también se incluyen utilidades de logging, (diarios escritos) muy poderosas, y facilidades de control de acceso que se puede aplicar en base a por dominio en el BIND 8. Este archivo de configuración efectúa la misma función que las del archivo boot en el BIND 4.

Archivos de Data (Archivos Maestros)

Aquí le presentamos un ejemplo de un archivo de data para la zona abiertos.org del servidor primario:

```
;  ejemplo de archivo de data del servidor de nombre para la zona  abiertos.org y la subzone descarga.abiertos.org.
STTL          86400
abiertos.org.         IN      SOA    sede.abiertos.org.       backup.total.abiertos.org.     (
                                                              53                    : Número de version
                                                              10800                 : Refrecar después de 3  horas
                                                              3600                  : Reintentar después de 1 hora
                                                              432000                : Expirar después de 1 Semana
                                                              10800   )             : TTL Negativo  de 3  horas
abiertos.org.                 IN     NS                       sede.abiertos.org.
abiertos.org.                 IN     NS                       gis.abiertos.org.
sede.abiertos.org.            IN     A                        200.42.210.43
gis.abiertos.org.             IN     A                        200.42.210.44

descarga.abiertos.org.        IN     NS                       parcial.sede.abiertos.org.
sede.descarga.abiertos.org. IN       A                        200.42.210.45
```

El formato del archivo de data de la zona es casi identico entre las versiones de BIND 4 y 8, como son todos los aspectos de configuración del DNS.

Los recursos listados en este archivo de record indican lo siguiente:

- Los records listados no deben permanecer en cache por más de 86,400 segundos (24 horas) al menos que sea especifícado.
- El host sede.abiertos.org tiene autoridad sobre la zona de abiertos.org.
- Hay dos servidores de nombre para la zona de abiertos.org: sede.abiertos.org y gis.abiertos.org.
- Las direcciones de los servidores de nombre.
- El host sede.abiertos.org es un servidor de nombre para la zona descarga.abiertos.org (delegamos la administración de la zona).
- La dirección de ese servidor de nombre.

El formato del archivo de zona del BIND requiere que los FQDNs, esos especificando el dominio completo hasta la raíz, deben ser escritos con un punto al final. En realidad se está especificando el root, ya que el nombre de root es "", el caracter null.

En la resolución de nombre, el record NS refiere el requiriente a otro servidor de nombre. Ha sabienda que el nombre del servidor de nombre no es suficiente. Para comunicarse con el, necesitaremos la dirección IP del servidor de nombre. Esta información es proveída como un record de Address cuando especificamos un servidor de nombre que va ha ser consultado. El servidor de nombre lo retorna como parte de su respuesta al efectuar referencia a otro servidor de nombre. Estos records de dirección proveen el mecanismo que permiten que los DNS funcionen correctamente.

Configurar un Servidor de Nombre

Para poder configurar un servidor de nombre en un sistema GNU/Linux, deberá efectuar los siguinetes pasos:

1. Crear el archivo maestro que contine los records de los recursos perteniente a su dominio. Estos incluyen:
    - La información de nombre a dirección del mapeado de los hosts en su dominio
    - El mapeado de dirección a nombre para los hosts en su dominio
    - Archivo cache que contiene la información referente al servidor root
    - El mapeado inverso de dirección para su loopback local (127.0.0.1)
2. Configurar un resolutor para su host para que las aplicaciones en su sistema usen el daemon del servidor de nombre.
3. Configurar su archivo /etc/named.conf con la información acerca de la localización y autoridad de los

archivos maestros.

4. Iniciar el daemon del servidor de nombre, named, y verificar que no cometió ningún error.

Información de Depurar

El daemon named proveé información atraves de mensajes que envia al syslog. Su PID se almacena en /var/run/named.pid . Para encender la opción de depurar/debugging, inicie named con la opción -d y enviele una señal:

kill -USR1 'cat /var/run/named.pid'

Para incrementar aún más el nivel de depuración les enviamos otros USR1. Para apagar por completo la depuración, efectué el siguiente comando:

kill -USR2 'cat /var/run/named.pid'

Existen varias tecnicas para ver y diagnosticar porque ciertos aspectos de su servidor de nombre no este funcionando apropiadamente:

- Examinar los mensajes del log por lo general se pueden accesar con tail /var/log/messages.
- Volcar la base de datos del servidor enviandole a proceso in.named una señal INT. Este volcado se escribe a un archivo nombrado named_dump.db en el direcotorio raíz del servidor de nombre o al directorio /var/tmp/, dependiendo de la versión de BIND que este usando. Para volcar la base de dato actual y el cache al disco ejecute el siguiente comando:
  kill -INT 'cat. /var/run/named.pid'
- Ejecute in.named en modo de depuración/debug usadno la opción -d. La salida es escrita a un archivo de nombre named.run, como anteriormente o al direcotorio root del servidor de nombre o al directorio /var/tmp.
- Incrementar el nivel de depuración del servidor ya ejecutándose enviandole la señal USR1. Esta salida es escrita al archivo named.run. Mientras más elevamos el nivel de debug, más mensajes y la duración del mensaje se torna. Para detener el modo de debugging por completo, se le envia la señal USR2.

Ejercicio 6-1: Configurar un Servidor DNS

Escriba los archivos de configuración del servidor de nombre en un subdirectorio de su home. Necesitará el la ruta completa a este directorio para colocarla luego en lugar de (nombre_de_Directorio). No se proveen soluciones para estos ejercicios.

1. Escriba el archivo de arranque/boot, named.conf, para su dominio:

```
# BIND v8 named.conf para el servidor primario de la zona  Dominio
# opciones del servidor
options {
        directory   "nombre_de_Directorio";
};
# zonas
zone  "."   in  {
        type  hint;
        file "root.cache"
};
zone  "0.0.127.in-addr.arpa"  in  {
        type master;
        file "db.127.0.0";
};
zone  "Dominio"  in  {
        type master;
        file  "db.Dominio";
};
```

2. Escriba el significado de cada línea menos las secciones que son comentarios.

Ejercicio 6-2: Archivo de cache del Servidor Raíz/Root

A los archivos de data le podemos dar cualquier nombre, siempre y cuando el archivo de configuración named.conf se refiere a el por ese nombre. Soluciones a este ejercicio se encuentran en el Apéndice B.

1. Cree el Servidor Raíz/Root Server, archivo será root.cache:

```
;   root.cache
;  pre-cargar el servidor primario con la información
;   a cerca de los sservidores root
                          99999999      IN      NSRootSrv.
RootSrv.                  99999999      IN      A       RootIP
```

Tome en cuenta que el RootSrv debe ser el FQDN del servidor root (como es ns.abiertos.gov.).

2. ¿Por qué necesitamos un record A (Address) para el servidor root?

Ejercicio 6-3: Archivo Data de los Hosts (Opcional)

No se proveen soluciones a este ejercicio.

Escriba el archivo data de los hosts para su subdominio, web.Dominio.org:

```
;   información del host para la zona Dominio.
;
$TTL          86400
Dominio.     IN    SQA SrvPrimario.     postmaster.SrvPrimario.
                                    1;   Serial
                                    10800;   Refrescar después de 3  horas
                                    3600;     Reintento después de 1  hora
                                    604300;   Expira después de 1 semana
                                    86400;  TTL Minimo de 1 día
Dominio.                           IN  MS  SrvPrimario.
SrvPrimario                        IN  A    IPPrimario
;  información de los hosts individuales...
host1.Dominio.                     IN A IP_host
```

Si se le dificulta entenderlo, hechemos un vistazo al siguiente ejemplo:

```
; información del host para la zona xyz.edu
$TTL                  86400
xyz.edu.              IN  SOA      escuela.xyz.edu.       postmaster.xyz.edu.   (
                                                 1;          Serial
                                                 10800;   Refrescar después de 3 horas
                                                 3600:     Reintentar después de 1 hora
                                                 604800: Expira después de 1 semana
                                                 86400):  Minimo TTL de 1 día

xyz.edu.                IN        NS     escuela.xyz.edu.
escuela.xyy.edu.        IN        A            200.42.200.43
; Información de los hosts individuales
pop.xyz.edu.            IN        A            200.42.200.47
matematica.xyz.edu.     IN        A            200.42.200.48
fisica.xyz.edu.         IN        A            200.42.200.49
quimica.xyz.edu.        IN        A            200.42.200.50
```

El record SOA (Start of Authority) indica que el host escuela.xyz.edu tiene autoridad sobre la zona xyz.edu. Es un record de recurso único, que se define sobre unas cuantas líneas usando parentesis. También

específica la dirección de correo del administrador responsable de la zona (postmaster@xyz.edu), con la @ reemplazada con un punto.

El record NS lista los servidores para la zona xyz.edu; estamos usando solamente un servidor. El record A es un record glue. Sin este record los servidores de nombre supieran el nombre del servidor autorizado, pero no su dirección, y por esto no pudiesen comunicarse con el.

Finalmente, hay records del tipo Address para cada host en la zona; esta es la información que estamos realmente interesados y es la razón principal de la existencia del DNS.

Ejercicio 6-4: Archivo del Host Local (Opcional)

No se proveen soluciones para este ejercicio.

1. Escriba el archivo del host local , backup.127.0.0.

El archivo del host local mapea la dirección de la interface loopback local (la cual normalmente es 127.0.0.1) al nombre "localhost". Recuerde que el record SOA (incluyendo el postmaster) todos deben estar en una misma línea:

```
$TTL        86400
0.0.127.IN-ADDR.ARPA.      IN  SOA   SrvPrimario.   postmaster.SrvPrimario.    (
                          1;  Serial
                          10800;   Refresca después de 3 horas
                          3600;  Reintenta después de 1 hora
                          604800;  Expira después de 1 semana
                          86400;   Minimo TTL de 1 día
                          )

0.0.127.IN-ADDR.ARPA.     IN   NS     SrvPrimario.
SrvPrimario.              IN   A       IPPrimario

1.0.0.127.IN-ADDR.ARPA.   IN   PTR     localhost.
```

2. Iniciar el servidor de nombre:

```
# cd   /etc/rc.d/init.d
# named start
```

Revise el archivo log (/var/log/messages) para ver si el servidor de nombre si inicio correctamente o si hubiesen errores en sus archivos de configuración.

ARCHIVOS DE RESOLUCION DE NOMBRE Y COMANDOS

En esta sección, revisaremos paso a paso varios aspectos de la configuración y los comandos del cliente DNS. En esta sección cubriremos los aspectos practicos de la configuración del software cliente para tomar las ventajas de un servidor. En esta sección también cubriremos los comandos usados para interactuar con la base de datos del DNS para recibir la inforrmación y diagnosticar y reparar servidores DNS cuando presentan problemas.

Los siguientes tópicos son discutidos en esta sección:

- Resolución de Nombre
- /etc/hosts
- /etc/resolv.conf
- Orden de Consulta
- /etc/nsswitch.conf

- /etc/host.conf
- Herramientas y Resolución de Problema

Resolución de Nombre

Para un cliente poder conectarse a un host remoto, este debe convertir el nombre del host a una dirección IP. La resolución de nombre es la habilidad de un ordenador de interpretar un nombre de dominio como es www.dominio.org y determinar su dirección IP. La libreria resolutora en el cliente efectuá esta tarea. Existe un número de archivos de configuración que controlan la manera en que funciona la libreria resolutora.

/etc/hosts	Este archivo contiene los hosts que no están listados en el DNS o que prefieren resolver sin el uso de un servidor de nombre. Este archivos es también para los clientes sin acceso a un servidor de nombre.
/etc/resolv.conf	Este contiene la lista de los servidores de nombres a consultar. Este puede contener más de uno; ellos serán consultados en orden. Si no hay ninguno presente, se consultará el localhost.
/etc/host.conf	Junto con archivo nsswitch.conf, este archivo determina cual método de resolución de nombre es usado primero usando el archivos hosts del cliente, un archivo hosts suplido por NIS o un servidor de nombre.
/etc/nsswitch.conf	Junto con host.conf, este archivo determina cual método de resolución de nombre es usado primero cuando se usa el archivo hosts del cliente, un archivo hosts suplido por NIS o un servidor de nombre.

/etc/hosts

El archivo hosts provee la información pertinente a los hosts de Internet. El archivo hosts se encuentra en el directorio /etc en las distribuciones de GNU/Linux y es un archivo de texto plano que contiene las direcciones IP y los nombres de host de los clientes y servidores en la red interna y hasta en el Internet.

El contenido del archivo por lo general es parecido a la siguiente entrada:

127.0.0.1 miguel.midominio.org miguel

El formato del archivo /etc/hosts es listar las direcciones IP primero, luego el número de nodo, seguido por el nombre alias. Se pueden incluir todos los alias necesarios. Si sólo un ordenador será agregado a este archivo, usted sólo necesitará la entrada de dirección loopback. Este es llamado el localhost.

El archivo hosts por lo general sólo contioene unas cuantas simple entradas, que incluyen el nombre y la dirección del host local. Esto es para permitir la dirección IP local que se encontrará cuando no se encuentra ejecutando un servidor de nombre, como por ejemplo cuando el ordenador se esta iniciando/booting (puede que ifconfig use el archivo hosts para determinar la dirección IP local).

También se puede suplir un archivo hosts via NIS (Network Information Service). NIS es un sistema de administración de redes que puede ser configurado para suplir archivos suplementarios a un sistema UNIX, incluyendo el archivo hosts. Usando NIS, un administrador de red puede distribuir un archivo de hosts único a múltiples clientes, asi proveiendo un servicio de resolución de nombre basado en la LAN sin el uso de un servidor de nombre.

El archivo hosts ya no es usado ampliamentre debido a que los servidores de nombre son muy comunes y son mucho más fácil de administrar que un archivos en cada cliente individual.

/etc/resolv.conf

Si el cliente no pudo resolver el nombre del host via el archivo hosts o no ha sido configurado para usar el archivo hosts, el próximo sistema lógico a resolverle su un nombre de host sería el servidor de nom-

bre. El archivos /etc/resolv.conf le dice a los rdenadores la localidad del servidor de nombre a consultar. El contenido del archivo /etc/resolv.conf por lo general luce como este que sigue:

Lo que sigue es un ejemplo del archivo resolutor para la zona de un cliente en la zona abiertos.org:

```
; resolv.conf para el cliente abiertos.org
domain          abiertos.org
search          abiertos.org
nameserver  200.42.200.42  ;  backup.abiertos.org
```

Hay tres items principales de información un resolutor necesita saber. El primero es el dominio por defecto- el dominio considerado local al host. El segundo es el dominio por defecto para agregarle a los nombres de hosts para que ellos asi no tener que especificar el FQDN para los hosts dentro de su dominio. La tercera entrada es la dirección del servidor de nombre del host local.

El archivo resolv.conf arriba mostrado indica que el dominio por defecto para este host es abiertos.org y el servidor de nombre que lo consulte debe ser enviado a la dirección 200.42.200.42 .

El archivo puede contener más entradas para que el servidor de nombre pueda consultar (por lo general alrededor de tres).

Orden de Consulta

El orden por el cual el resolutor consulta el DNS y el archivo local /etc/hosts o la desición si de hecho revisa el archivo local depende de la implementación de la configuración de la implementación del resolutor y de como los archivos de configuración locales están configurados. Los detalles de estas implementaciones no están definidas e la documentación de Internet de los RFC. El resolutor puede:

- Consultarprimero el DNS, luego revisar el archivo /etc/hosts si se retorna un error.
- No usar de ninguna manera el archivo /etc/hosts.
- Permitir que se efectue una elección via un archivo de configuración.

Este puede depender si se encuentra un NIS ejecutándose en el ordenador local. Por ejemplo, Sun Solaris 1.x no utilizaba el archivo /etc/hosts para nada y primero consultaba NIS.

Los sistemas GNU/Linux permiten que se efectue una elección via los archivos /etc/nsswitch.conf y /etc/host.conf.

/etc/nsswitch.conf

Un archivo de nombre /etc/nsswitch.conf determina el orden que las busquedas se efectuan cuando se requiere cierto tipo de información. El archivo nsswhch.conf es una base de datos del sistema. Este archivo también se encuentra en el directorio /etc. Se agrega una entrada en el archivo /etc/nsswitch.conf para cada base de datos. Por lo general estas entradas son muy simple, como por ejemplo “protocols: files” o “networks: files nisplus”. Pero, cuando son especificada múltiples fuentes, hay veces que es necesario definir precisamente las curcunstancias bajo las cuales cada archivo fuente será tratado. Los archivos deben ser ordenados con el servicio más usado en el principio.

Entradas válidas en el archivo incluyen:

nisplus o nis+	Usar NIS+ (NIS versión 3)
nis o yp	Usar NIS (NIS versión 2), también conocido como YP
dns	Usar DNS
files	Usar los archivos locales

db	Usar los archivos de la base de datos local (.db)
compat	Usar NIS en modo compacto
hesiod	Usar Hesiod para la búsqueda de usuarios
[NOTFOUND=return]	Detener búsqueda si aún no se encuentra

El contenido típico del archivo nsswitch.conf incluye:

passwd:	files nisplus nis
shadow:	files nisplus nis
group:	files nisplus nis

La primera columna es la base de datos y el resto de la línea específica como el proceso de búsqueda funciona. La especificación de la configuración para cada base de datos pueden que contenga dos cosas:
- Cuestiones específicas a servicios como “files” o “nis”
- La reacción del resultado de una búsqueda como “[NOTFOUND=return]”

En el caso de la resolución de nombre, una entrada puede ser asi:

hosts: files nisplus nis dns

La línea anterior primero consultaria el archivo hosts local, luego el servidor de NIS+, seguido por el servidor NIS y por último el DNS.

/etc/host.conf

El archivo host.conf contiene la información de configuración acerca de la libreria resolutor. El contenido debe tener sólo una palabra claver por línea con la información de configuración justamente al lado.

order	Esta palabra clave específica como la búsqueda de host deben ser efectuadas. Este también tiene por lo menos un método de búsqueda, separado por comas si hay más de uno. Estos métodos son bind, hosts y nis.
trim	Esta puede estar listada multiple veces. Debe ser seguida por un único nombre de dominio, con un punto al final. Cuando establecida, la libreria resolv+ automáticamente recortará/trim el nombre de dominio dado del final de cualquier nombre de host resuelto via DNS. Esta es para ser usada con hosts locales y dominios.
multi	Los valores de esta palabra clave son on y off. Si establecida a on, la libreria resolutora retornará una dirección válida para que un host aparesca en el archivo /etc/hosts. El valor por defecto es off, ya que tiende ha ser más lento en sitios con archivos hosts amplios.
nospoof	Los valores de esta palabra reservada son on y off. Si esta opción se establece a on, la librería resolv+ tratará prevenir que el spoofing/fantasma de nombre de hos para mejorar el nivel de seguridad de rlogin y rsh. Luego de ejecutar una búsqueda de una dirección de un host, resolv+ efectuará una búsqueda de un nombre de host para esa dirección. Si dos nombres de host no se igualan, la consulta fracasará.
alert	Los valores de esta palabra clave son on y off. Si esta opción esta habilitada y la opción nospoof tambieen esta establecida, el resolv+ colocaría en el log una entrada de advertencia/warning del error via el utilitaro syslog. El valor por defecto es off.
reorder	Los valores de esta palabra clave son on y off. Si esta opción se establece en on, resolv+ intentará reordenar las direcciones host para que las direcciones locales esten listadas primeras cuando se ejecute un gethostbyname. El reordenamiento se efectua para todos los métodos de búsqueda. El valor por defecto es off.

Un archivo /etc/host.conf típico luce asi:
order hosts.bind
multi on

La línea con la orden dice que primero busque en el archivo /etc/hosts antes de buscar el nombre del host usando un DNS. El BIND es el servidor DNS más usado y una parte de este es usada en los sistemas GNU/Linux para efectuar resolución de nombre en el lado del cliente. La palabra multi en la línea permite que líneas en el archivo /etc/hosts tengan más de una dirección IP. En otras palabras, este se mantiene buscando en el archivo hosts aún después de haber encontrado el mapeado correcto para ver si existen otros mapeos que también apliquen.

Herramientas y Resolución de Problemas

Ambos los administradores de sistemas y los usuarios a veces se encuentran tratando de diagnosticar y corregir un problema relacionado con la resolución de nombre, muchas veces sin percatarse de la relación. Hay muchas herramientas disponible para la corrección de problemas relacionados con resolución de nombre asi como para investigar información de domios, entre las cuales se incluyen:

- nslookup
- whois
- nslint
- dig

nslookup

El comando nslookup cuestionará al servidor de nombres especificado en el archivo /etc/resolv.conf acerca de la máquina especificada y devolvera su dirección IP. El formato de este comando es: nslookup máquina

En modo interactivo, nslookup puede hacer mucho más que sólo encontrar direcciones IP: puede preguntarle al servidor de nombres por cualquier clase de registros (no sólo A) e incluso puede mostrar la información referente a una zona entera.

Para entrar en el modo interactivo sólo es necesario teclear nslookup. El programa contestará con un signo de menor que ``>'' indicando que está listo para ejecutar comandos. Es posible entonces indicarle cualquier nombre de dominio y nslookup buscará por registros de tipo A. Para cambiar el tipo de registro que queremos encontrar es posible indicar set type=tipo, donde tipo puede ser cualquiera de los que ya mencionamos en secciones anteriores o incluso any, que indica cualquier tipo de registro.

La siguiente sesión, que se incluye como ejemplo, muestra cómo es posible encontrar información no sólo acerca de máquinas (direcciones IP), sino acerca de dominios (cuáles son sus servidores de nombres o de correo).

nslookup nos permite consultar interactivamente a un servidor de nombre. Distribuido con el BIND, debe estar disponble en todas las plataformas. Este comando se comunica con un servidor a la vez, ejecutando peticiones de un tipo particular de records de recursos desde un dominio especificado. Este no utiliza NIS o una base de datos de host.

Lo siguiente es un ejemplo del comando nslookup:
[root@proxy-ap root]# nslookup -sil abiertos.org
Server: 10.0.0.1
Address: 10.0.0.1#53
Non-authoritative answer:

Name: abiertos.org
Address: 206.123.67.148

Estos utilitarios son extensos y deben ser estudiados a fondo para poder dominarlos por completo.

whois

El comando whois es una herramienta útil para autentificar las cuentas de usuario y adquirir información sobre el servidor, incluyendo información de facturacion y un soporte telefonico. El comando alinia la sintaxis como sigue:

whois nombre_servidor@whois_server

El comando whois típicamente es usdo para identificar los usuarios, para locaizar nombres de dominios libre o hasta para revisar los records de un servidor DNS. Por ejemplo, si necesitamos saber el nombre de host de un usuario, el administrador puede revisar su servidor y si es necesario, llamar y verificar la cuenta. Un administrador también puede encontrar útil para verificar si un nombre de dominio que el desea utilizar aún esta disponible o no.

Existen unos cuantos servidores whois disponibles. Compañías, organizaciones y universidades todas deben registrar sus nombres de dominios con un servicio de registro de nombre (Domain Name Registration Service). Estas compañías registran y venden nombres de dominios. Ellos también mantienen una base de datos de todos los clientes que con ellos han registrado un dominio y los despliegan en su servidor whois. Como hay tantos servidores whois que solo contienen la información de los clientes de sus organizaciones, consultar el servidor para saber en cual servidor whois es que esta listado. Los dos servidores más amplios del Internet son whois.register.com y whois.networksolutions.com.

Red Hat utiliza una distribución de whois llamada fwhois. El fwhois no ofrece todas las opciones que el servidor whois provee. La gran mayoría de servidores whois permiten que palabras reservadas (como son Domain, Registrar, etc.) sean usadas para registringir la información retornada a campos especificos. De esta forma, es posible buscar un servidor por su nombre de dominio o buscar un servidor y sólo desplegar el nombre de dominio. Excisten un cliente para el X Window de nombre xwhois el cual es un cliente gráfico que despliega información en una ventana X Window y provee un interface de manejo intiutivo para usuarios sin experiencia con los servidores whois.

nslint

La herramienta nslint es útil para los administradores de servidores de nonbre ya que esta los asiste en identificar errores en los archivos de zonas. El uso del comando es realmente simple. Si su archivo named.conf se encuentra en /etc y las rutas están establecidas correctamente, sólo tendrá que ejecutar el comando sin ninguna opciones:

nslint

Esta procederá a reportar cualquier error a pantalla. Si recibe un gran número de errores, puede que le sea más fácil direccionarlo por tubería a less o more para poder desplegarlo una página a la vez:

nslint |less o # nslint |more

Si prefiere almacenar la informacióna un archivo log, podrá entonces direccionar la salida a un archivo:

nslint > ns_erorres.txt

La herramienta nslint tiene varias opciones para afectar el compartamiento de su salida. Dos de las más usadas son:

-c Específica un archivo named.conf diferente al por defecto en el directorio /etc
-d Desplega más información de las cosas a revisar

Si su distribución no incluye la herramienta nslint, podrá descargarla desde el Internet.

dig

La herramienta dig es similar al nslookup. De hecho nslookup está deprecada y devemos usar dig como su reemplazo. La herramienta dig es usada para buscar información directamente desde los servidores de dominio. esta puede efectuar simple busquedas de DNS o ser usada para ayudar a diagnosticar y solucionar problemas de DNS.

La herramienta dig tiende a proveer más información que otras herramients de búsqueda de infomación de dominio. ejecutar dig con abiertos.org como argumento nos desplega una página de información:

```
$ dig abiertos.org

; <<>> DiG 9.2.2 <<>> abiertos.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12082
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;abiertos.org.                  IN      A

;; ANSWER SECTION:
abiertos.org.           4946    IN      A       206.123.67.148

;; AUTHORITY SECTION:
abiertos.org.           4639    IN      NS      ns1.quxnet.net.
abiertos.org.           4639    IN      NS      ns2.quxnet.net.

;; ADDITIONAL SECTION:
ns1.quxnet.net.         157352  IN      A       206.123.67.145
ns2.quxnet.net.         157352  IN      A       206.123.67.146

;; Query time: 512 msec
;; SERVER: 10.0.0.1#53(10.0.0.1)
;; WHEN: Wed Jun  8 14:56:05 2005
;; MSG SIZE  rcvd: 124
```

Si comparamos con la salida de nslookup de abiertos.org notamos que mucho menor:

```
$ nslookup  -sil abiertos.org
Server:         10.0.0.1
Address:        10.0.0.1#53

Non-authoritative answer:
Name:   abiertos.org
Address: 206.123.67.148
```

RESUMEN

En este capítulo, cubrimos el Sistema de Nombre de Dominio (Domain Name System, DNS). Entre los puntos principales que tocamos se incluyen:

- Un servicio de nombre que traduce nombres de host a direcciones IP.
- El TCP/IP utiliza a DNS como su servicio de nombre.
- El DNS usa unsistema de dominio jerarquico donde cada dominio puede contener sub-dominios.
- La administración de los dominios de DNS es delegada a base de datos (llamadas zonas) que son distribuidas en todo el Internet.
- Los dominios pueden contener múltiples zonas.
- Cada servidor mantiene los records de recursos de su zona.
- Al configurar un servidor DNS, deberá configurar archivos maestros.
- Se recomienda utilizar herramientas administrativas para asistirle en escribir sus archivos maestros
- Existen otros servicios de nombre que son similares al DNS incluyedo el NIS de Sun entre otros.

PREGUNTAS POST-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Son las siguientes acertaciones ciertas o verdaderas?

   • Cada host debe ejecutar un servidor de nombre individual

   • Un servidor de nombre puede ser primario a dos zonas.

   • Un host debe saber la localidad de un servidor de nombre de su zona padre

   • Un servidor de nombre debe servir por lo menos a una zona.

2. ¿Por qué es que alguna data es no autorizada (data nonauthoritative)?

3- Nombre un servicio de nombre alternativo.

4. Nombre tres tipos de records de recursos.

5. Hay dos convenciones usadas al elegir un dominio. ¿Cuáles son estas dos convenciones? Nombre dos ejemplos de cada una.

OTROS SERVICIOS DE RED

OBJETIVOS

Al completar este capítulo, usted será cápaz de:

- Configurar un sistema GNU/Linux como un cliente o servidor DHCP.
- Configurar y usar un servidor anonimo FTP.
- Configurar un servidor telnet y explique poque es deseable deshabilitarla.
- Instalar un servidor Squid para habilitar el cache de páginas Web.
- Explicar como funciona un servidor de noticias (news).
- Habilitar la sincronización usando NTP.
- Configurar un cliente y un servidor NIS.
- Instalar y Configurar un servidor LDAP.

PREGUNTA PRE-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. Nombre tres clientes FTP.

2. ¿Qué significan las siglas LDAP y que hace este?

3. ¿Cuál servicio de red provee direcciones IP?

4. ¿Cuál es la diferencia entre NTP, ntpd, y xntpd?

INTRODUCCION

Una de las fortalezas de GNU/Linux es su habilidad de proveer una variedad de servicios de red. En este capítulo, cubriremos varios servicios que tarde o temprano usted necesitará implementar, entre estos se incluyen DHCP, telnet, FTP, Squid, NIS, y LDAP.

Los primeros dos servicios que se introducirán son servicios que proveen control y cierto niveles de seguridad para otros servicios:

- inetd El daemon que administra los otros servicios
- tcpd Un daemon de seguridad que provee control del acceso rudimentario

INETD y XINETD

A menudo referido como el Super Servidor o el Super Daemon de Internet porque de las muchas funciones que este puede efectuar, el inetd es una parte importante del sistema porque este ayuda a conservar recursos, actua como una primera línea de defensa contra los usuarios no deseados y puede restringir acceso al root por usuarios permitibles. Solo se puede encontrar un inetd a la vez y por lo general se inicia cuando el sistema primero se inicia desde los scripts locales en el directorio de control /etc/rc.local.

El inetd puede efectuar servicios internos simple como son echo, discard, generador de caracteres (chargen), daytime y time. Daytime es el tiempo presentado en un formato entendible por humanos. Time es el tiempo en un formato presentado para las máquinas en la forma del número de segundos desde la medianoche de, Enero 1, 1900. El inetd provee ambos servidores basados en TCP y UDP para estos servicios asi como para todos los otros.

Una de las funciones principales del inetd es que este puede actuar como un vigilante para los sockets (zocalos) Internet. El inetd monitoreá un número de sockets diferentes esperando que un usuario externo efectué un contacto. Una vez una conexión es hecha, el inetd decide a cual servicio el socket corresponde y lanza (spawns) un programa que puede manejar la petición.

El inetd puede manejar tramas y servicios de tipos datagramas. Si la petición es de servicios de tramas, después que inetd ha pasado el socket al servidor, regresara y escuchara por nuevas peticiones de coneccion. Si las petición es para servicios de datagramas, después que inetd ha pasado el socket al servidor, este tiene que esperar, ignorando la actividad del socket de datagramas hasta que el servidor exista. La opción de esperar puede ser utilizada para tramas y/o servidores multihilo solo debe ser cambiado en el archivo de configuración inetd.conf.

Otra función principal de inetd es que este aligera la carga sobre el sistema, inetd es sólo un demonio, que puede invocar a otros demonios cuando es necesario en vez de usar a un número de diferentes demonios permanentes y aislados. Esto libera recursos, por consiguiente disminuye la carga sobre el sistema,

Hay algunas desventajas a la utilización inetd a diferencia de la utilización de demonios aislados. Correr inetd quiere decir que siempre que una petición de conexión entre, inetd debe realizar un número de procesos diferentes para manejarlo. El inetd debe bifurcar un nuevo proceso y luego cargar al nuevo demonio. Por consiguiente, el nuevo demonio debe cargar y analizar (dividir el código sobre las más pequeñas partes que pueden ser analizadas) su configuración. Todos esto combinado crea un proceso más largo que si usted usara a demonios independientes. Cuando se usan demonios independientes, ellos reciben una petición, hacen una copia de sí mismo y la copia maneja la petición.

A pesar de esta desventaja, usted probablemente querrá usar inetd. La ventaja de usar inetd sobre pasan sus pequeña desventaja. El uso de demonios independientes desperdician muchos recursos del sistema, recur-

sos que los servidores pequeños no disponen para malgastar, inetd es generalmente un programa bueno y es una parte importante del sistema Linux.

Hoy en día existen dos demonios populares de Internet usados por distribuciones GNU/Linux:
- inetd
- xinetd

El inetd

Por muchos años, inetd ha sido el demonio de Internet usado por distribuciones GNU/Linux. Este usa un solo archivo de configuración, /etc/inetd.conf. Dentro del archivo/etc/inetd.conf, los servicios controlados por inetd son catalogados, uno por línea. En cada línea, hay siete campos. Para cada servicio configurado, debe haber una entrada en los siete campos. Los campos son así:

service_name socket_type protocol wait/nowait user server command line

service name	Este es traducido a números de puertos buscandolos en el archivo de servicios(/etc/services) para los servicios TCP y UDP o el demonio portmap para servicios RPC.
socket type	Este debe ser el tipo de socket que el servicio usaria. Esto va a ser stream, dgram, raw, rdm, o seqpacket. Los servicios basados en TCP siempre deberían usar el tipo stream, mientras los servicios basados en UDP deberían usar dgram.
protocol	Este debe ser un protocolo valido encontrado en /etc/protocols. TCPMUX debe usar TCP. Normalmente TCP o UDP es usado.
Wait/nowait	Este le dice al servicio si debe procesar peticiones múltiples juntsd o no. El datagrama (dgram) debe esperar, mientras que el stream o el socket multihilo deberan usar nowait.
User	Este debe contener el usuario que el servidor debera ejecutar. Esto es como un servidor puede usar inetd para limitar el acceso con el root. Esto es hecho simplemente cambiando nombre del campo de usuario por un usuario de clase inferior o un usuario con menos acceso a diferencia de root. Un ejemplo de esto sería si el nombre fuera cambiado por invitado (guest).
server	Este debe contener el nombre de la ruta del program a a ejecutar por inetd sobre la conexión. Si el servicio es interno, entonces este campo simplemente debería ser “interno”.
command_line	Esto contendrá una línea de comandos o argumentos para inetd, Este campo debería comenzar con el nombre corto del programa, argv [0]; sin embargo, probablemente es ocultado por el shell. Otra vez, si los servicios son internos, entonces el campo debería leer “interno”.

Por ejemplo, es asi como los campos deben aparecer en el servico Ftp:

ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -1 -a

Siempre que el archivo de configuración de inetd sea cambiado, el demonio tiene que releer el archivo antes de que los cambios surtan efecto. Las opciones siguientes permitirán a inetd para releer el archivo:

-c	Este fuerza inetd a releer el archivo de configuración. Este envía la señal de-SIGHUP al inetd actualmente corriendo.
-k	Este comando mata el inetd ejecutándose actualmente. Este envía el comando de -SIGTERM a inetd, haciéndolo retirarse. Usted entonces debe reiniciar inetd.

Las opciones siguientes pueden ser usadas con inetd, si usted es el superusuario:

-d	Este habilita depuración (debugging).
-q	Esto establece el tamaño de la coloa/queue de espera del socket a un especificado valor. Por defecto es 128.
-R	Esta opción específica el número máximo de veces que un servicio puede ser invocado en un minuto. Por defecto es 1,000 por minuto.
-l	Este habilita o desabilita el log de conexión para inetd. Esto puede ser práctico porque inetd escribe al

log todas las conexiones acertada o fracasada. Si hay un registro de una negación repetida del mismo sistema remoto, hay posibilidades que esten tratando de rompers nuestro sistema. El comando -l envía el comando de SIGQUIT al inetd..

El xinetd

El otro demonio popular de Internet es xinetd. Hace ya un tiempo que xinetd a progresivamente a empezado a ganarse un espacio en las distribuciones de GNU/Linux. “ Desde el lanzamiento de Red Hat Linux versión 7, este es el demonio por defecto de Internet en una de las distribuciones más reconocidas en el mercado de GNU/Linux.

El formato de archivo de configuración para xinetd es diferente de él tradicional /etc/ inetd. Confudido por inetd los usuarios de xinetd usa una configuración de dos etapas: el archivo de configuración para el servidor xinetd sí mismo y un directorio que contiene los archivos de configuración individuales para cada uno de los servicios bajo su control.

Un archivo de configuración de /etc/xinetd.conf puede lucir así:

```
# archivo de configuración simple para xinetd
# Algunos por defecto, y incluye /etc/xinetd.d/
defaults
{
instances               = 60
log type                = SYSLOG authpriv
log_on_success          = HOST PID
log_on_failure          = HOST RECORD
}
includedir /etc/xinetd.d
```

Un archivo de ejemplo de configuración puede ser este de un ftp:

```
# default:   on
# description; The wu-ftpd FTP server serves FTP connections.  It \
# uses normal, unencrypted usernames and passwords forauthentication,
service ftp
(
socket_type                    = stream
wait                           = no
user                           = root
server                         = /usr/sbin/in.ftpd
server_args                    = -1 -a
log_on_success                 +== DURATION USERID
log_on_failure                 += USERID
nice                           = 10 1
```

Muchos de los parámetros listados son similares a aquellos en el archivo /etc/inetd.conf, como son socket_type, user, server, y otros. El campo commandline de /etc/inetd.conf está presente como el campo de server_args en un archivo de configuración de servicio de xinetd.

Ejercicio 7-1: Configurar Servicios en xinetd

Con el lanzamiento de la versión 7 Red Hat GNU/Linux, xinetd fué incluído y presentado como un reemplazo para inetd. Mientras que la configuración de ambos presenta ciertas diferencias entre xinetd y inetd, ambos son lo relativamente intuitivos para que nos podamos facilmente adaptar entre el formato de uno y el otro. En este ejercicio, demostraremos como poner en práctica cambios a un par de servicios dentro de la nueva estructura.

La soluciones a este ejercicio se incluyen en su contenido.

1. Arranque su sistema en Linux.

2. Ingrese como root.

3. Verifique que el ftp se puede ejecutarse para el localhost. Una vez que usted ve que la conexión no falló, presione CONTROL+C para salir del shell y regresar al prompt.
ftp localhost
Connected to localhost.
220 localhost FTP server (Version wu-2.6.1(l) Ued Aug 9 05:54:50 EOT 2000)
ready.
530 Please login with USER and PASS.
Name (localhost:root): #

4. Ahora para deshabilitar el sendee, muévase al directorio del árbol de configuración del xinetd:
cd /etc/xinetd.d/

5. Aquí usted encontrará los archivos que son usados para controlar los servicios. Use un editor de texto para abrir el archivo de configuración de ftp:
vi wu-ftpd

6. Agregue las siguientes directivas de configuración:
disable = yes

Debe ser dentro de los corchetes. Un lugar bueno para ello sería después de la directiva.
r* 1 r*i

7. Guarde y cierre el archivo.

8. Reiniciar el super servidor xinetd:
#/etc/init.d/xinetd restart

9. Intentar conectarse por ftp al localhost otra vez. El servicio ahora debe estar cerrado.
ftp localhost
ftp: connect: Connection refused ftp> 10.

Salga del ftp escribiendo simplemente“bye” en el prompt del ftp.

El TCPD

El demonio tcpd es una manera simple de añadir cierto nivel de seguridad de acceso a aquellos servicios controlados por inetd. Esto es una envoltura o wrapper TCP, que intercepta peticiones y las envuelve desde su aplicación estandar a su propia. La función de tcpd es la de supervisar las conexiones al servidor hechas vía inetd y luego permitir o negar la conexión basada en la información contenida en los archivos hosts.allow y hosts.deny.

Para que tcpd trabaje, es necesario efectuarle cambios menor al archivo /etc/inetd.conf. Para cada servicio supervisado por tcpd, el listado de la ruta de las aplicaciones deberá ser cambiado a la ruta usado por tcpd. Por ejemplo, la entrada normal para el servicio de telnet en /etc/inetd.conf es:

Después del cambio, la entrada se convierte en la siguiente:
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd

This tricks inetd into running tcpd when a service is requested.

El tcpd entonces escribirá la petición al log y revisará las reglas de acceso en los archivos /etc/hosts.allow y /etc/hosts.deny, en ese orden, basado en el ID del usuario remoto. Una vez y tcpd encuentra la entrada aplicable para el host remoto y el servicio que este ha hecho la petición, el tcpd usa esa información para permitir o denegar la petición y no revisa los archivos más allá. Si no existe una una entra para el host ID en uno de los archivos hosts.allow o hosts.deny, entonces la conexión será permitida por defecto.

Los archivos hosts.allow y host.deny contienen un listado de los servicios y los hosts en un formato de servicio:nombre_de_host:comando, donde comando es un campo opcional. Un ejemplo típico del archivo hosts.allow es parecido al siguiente:

ALL: miguel ivelis dessi in.ftpd:
LOCAL EXCEPT jazzy in.telnetd:
192.168.2

La primera línea de este archivo hosts.allow nos dice que los hosts remoto miguel, ivelis y dessi tienen aaceso a todos los servicios ejecutándose en el host local. Las máquinas que se encuentran en el mismo dominio no tienen que presentarse con sus nombres de host completo. El uso de ALL en el campo servicio indica que cada servicio en el host está disponible a esos usuarios remotos.

La entrada ftp es muy interesante. La palabra LOCAL indica que todas las máquinas en el mismo dominio, esas que cuales sus nombres de host no están separadas por un “.”, tienen acceso al ftp. El uso de la palabra EXCEPT indica que la máquina de nombre jazzy en el dominio local no tiene acceso al ftp.

La última línea en este ejemplo describe los privilegios de acceso para el servicio de telnet. En este caso, es la subred 192.168.2 ha sido declarada especificamente con acceso al telnet.

Otra característica interesante del tcpd es la que cuando una entrada es encontrada en uno de los archivos hosts.allow o hosts.deny, un comando puede ser ejecutado después que el servicio permitido o denegado. Este asa el campo comando opción. El usuario puede colocar un comando en este campo y este se ejecutará siempre y cuando yna petición iguale esta entrada. Por ejemplo, se puede ejecutar finger para investigar quien desea acceder el servicio y los resultados pueden ser enviados por e-mail a root. Muy a menudo los ataques al sistema pueden ser evitados con simplemente efectuando un monitoreo de quien quiere accesar cuales partes del sistema.

DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

El DHCP es usado para asignar direcciones IP a los hosts de una red. Las asignaciones son administradas por un servidor centralizado, asi simplificando la administración. En esta sección discutiremos el protocol DHCP y dos de su antecesores, BootP y RARP. También cubriremos el uso y la configuración de los clientes y servidores DHCP sobre plataforma GNU/Linux.

Hay varias buenas razones para usar un DHCP. Una de las principales razones para el uso de un servidor DHCP es para que usted no tenga que configurar los valores de IP de cada máquina manualmente en su red. Este también elimina la necesidad de tener que contabilizar el IP individual asignada a cada máquina. También nos permite tener más máquinas o hosts que direcciones IP disponibles. Esto es muy útil para los ISP o comapñías con muchos usuarios, especialmente los que usan portátiles. Aunque podemos tener más hosts que direcciones IP, usted no podrá exceder el número de direcciones IP con los hosts conectados.

Los siguientes tópicos son discutidos en esta sección:
- Configuración Automatica vs. Dinámica
- RARP
- BootP
- DHP
- Configurar un Servidor DHCP
- Configurar un Cliente DHCP

Configuración Automática vs. Dinámica

Los parámetros del TCP/IP pueden ser configurados en varias maneras diferentes. En su origen, toda la información de configuración era estatica e introducida manualmente. Muchos istemas aún son configurados manualmente, especialmente los sistemas que son servidores. RARP y BootP introdujeron la configuración automática. El DHCP permite las dos opciones de configuración automática o dinámica.

En la configuración autmática, el cliente consulta a un servidor para recibir sus parámetros cuando primero se inicia y se une a la red. Estos parámetros, incluyendo la dirección IP, son los mismos todo el tiempo, al menos que el administrador efectúa un cambios en el servidor.

La Configuración Dinámica difiere de la configuración aumática en que al cliente se le asigna una dirección IP y configuración cada vez que se une a una red. A diferencia de la configuración automática, puede ser que se le asignen parámetros diferentes cada vez que inicie el sistema. La dirección IP es alquilada al cliente. El alquiler de una dirección IP expira después de un intervalo de tiempo especificado, luego del cual deberá ser renovado por el cliente o retornado al pool de direcciones y configuraciones. La Configuración Dinámica fué el cambio principal del BootP al DHCP.

La diferencia entre la configuración automática y dinámica es como el servidor determina la información de la configuración. Si este usa un pool de direcciones IP y escoje una al azar, entonces estamos utilizando una configuración dinámica. Si esta usa una tabla creada por el administrador de sistema para mapear direcciones MAC a direcciones IP, entonces estamos efectuando configuración automática. Ambos metódos utilizan un servidor para hacer la asignación actual a los clientes.

El DHCP ofrece tres diferente tipos de configuraciones:

Manual	Esta es equivalente al BootP.
Automática	El cliente es automáticamente asignado una dirección IP permanete y su configuración desde que se une a la red.
Dinámica	El cliente es asignado una dirección IP y su configuración cada vez que se une a la red. La dirección IP es alquilada al cliente.

En configuraciones automática y dinámica, el alquiler de la dirección IP (y la configuración del cliente) expiran después del intervalo de tiempo establecido, luego este cliente debe renovar su alquiler y la dirección será retornada al pool de direcciones y configuraciones del servidor.

RARP

El primer protocolo que fué usado para asignar direcciones IP automáticamente fué RAPR (Reverse Address Resolution Protocol). El RARP está definido en el RFC 903. Este se encuentra en la misma capa que el ARP y usa algunos de los mismos formatos de paquetes. Así que el RARP se encuentra al nivel inferior (o paralelo) al de la capa del protocolo IP. Esto tiene mucho sentido ya que es imposible enviarle un paquete IP a un cliente que aún no tiene una dirección IP asignada.

El protocolo RARP no usa IP, asi que este debe hacer interface directa con la capa de Enlace (Ethernet Data Link layer). Por esta razón es que el kernel debe implementar este protocolo. Al configurar el kernel Linux, usted notará dos entradas preguntandole si desea habilitar soporte de RARP: uno es para habilitar el sistema que funcione como un cliente de RARP y el otro es para que la máquina funcione como un servidor RARP. El servidor RARP contiene una tabla del mapeado de las direcciones MAC a direcciones IP. Si recordamos cuando cubrimos el ARP, esta tabla mapea las direcciones en forma inversa a la tabla de búsqueda del ARP.

Cuando un cliente de RARP se inicia y desea determinar su dirección IP, este crea un paquete de petición y envia un mensaje de difusión o broadcast atraves de la capa completa de Enlace de Data del Ethernet (Ethernet Data Link layer). Cuando el servidor RARP escucha la petición RARP, este busca la dirección MAC en su tabla de RARP. Si el servidor encuentra la dirección MAC en la tabla, este procede a enviar una respuesta RARP al cliente con la dirección IP que este encontró. Este envia esta respuesta RARP en un paquete raw de Ethernet directamente a la dirección MAC del cliente.

Una de las desventajas de RARP es que este requiere solamente un broadcast desde el cliente y una transmisión unicast desde el servidor al cliente. Pero, existen muchas limitaciones con el RARP, entre las cuales se incluyen:

- Para usar RARP sse requiere acceso directo a la capa de Enlace de Data. Esto hace que la programación de aplicaciones sea un poco más dificil y a menudo el protocolo termina siendo implementado en el kernel. Esta implementación es preferible en el espacio del usuario o el user space.
- Las respuestas RARP solo contienen una única dirección IP, no distribuyen ninguna información acerca de los routers, imagenes de boot del servidor, etc. Otro método es necesario pra proveer la información acerca de las imagenes de arranque y los otros parámetros para los clientes sin discos (diskless clients).
- Como se encuentra en la capa de Enlace de Data, RARP no se desplaza más allá de la red local. Por esto es que debe existir un servidor RARP en el mismo segmento local de cada cliente sin disco.

Para sobre pasar algunos de estos problemas, se desarrollaron estos protocolos de BootP y el DHCP.

BootP

Como el RARP, el protocolo BootP provee a los sitemas con direcciones IP que son automáticamente configuradas, pero provee más informacióny utiliza diferentes mecanismo. Además de informar sobre la dirección IP de un cliente, el BootP puede también proveer mucha más información. Algunos ejemplos son: la pasarela local por defecto (default gateway), direcciones de servidor de nombres, la dirección IP de servidor de imagen y el nombre del archivo de la imagen boot. El BootP fué descrito en el RFC 951 y fué disennado para ser usado por estaciones de trabajo sin discos. Para los clientes sin discos, el boot ROM obtiene la información del BootP y usa la información para contactar el servidor TFTP para cargar el sistema operativo.

Cuando un cliente BootP desea obtener una dirección IP, este envia un broadcasts de un paquete con un BOOTREQUEST a la subred local. Esto se logra usando la dirección IP de broadcast, 255.255.255.255. Por lo general la dirección IP será 0.0.0.0 en este paquete, indicando que el paquete proviene de un sisstema desconocido o uno sin una dirección IP válida. El broadcast es enviado al puerto UDP número 67. La respuesta es enviada al puerto UDP número 68, asi pues que el cliente debe escuchar en ese puerto por los paquetes de respuesta.

Al igual que el RARP, el servidor BootP busca en sus tablas para determinar a cual dirección IP enviar devuelta a los clientes. Este también puede contener información adicional que desea enviar de vuelta al cliente. Cada pedazo de información enviada es asociado con una etiqueta. La etiqueta describe el significdo

de la información. Las etiquetas son identificada en el documento RFC que describen el protocolo. El tipo de información que usted configura al servidor que efectué depende de las necesidades del sistema cliente involucrado. Por ejemplo, una terminal sin disco puede ser enviada la dirección IP de un servidor TFTP, mientras que a otro sistema le enviariamos otro tipo de información.

Para enviar el paquete de respuesta al cliente puede que resulte un poco complicado para el servidor ya que este debe enciar un paquete IP a un sistema sin dirección IP. Hay cuatros posibles métodos que pueden ser utilizados para enviar el paquete al cliente:

- Si el cliente requiere una dirección IP específica en el paquete de requiriente y el servidor concede al cliente usar esa dirección IP, la respuesta puede ser enviada directamente a la dirección IP del cliente.
- Enviar el paquete de respuesta vía un broadcast. La petición que el cliente envia contiene un identificar únco, asi que el cliente deberá escuchar en su puerto UDP número 68 por un paquete de respuesta que iguala su identificador.
- Usar la información del paquete que el cliente envia para forjar una entrada en la tabla de ARP del servidor. Entonces se puede enviar el paquete a la dirección IP y la pila de IP del servidor usará la entrada en la tabla ARP para mapear la dirección IP a su dirección MAC. El kernel Linux soporta este método. De hecho, se puede agregar una entrada a la tabla ARP manualmente usando el comando arp.
- Crea un paquete data-link por completo para enviar sobre el cable de la red. Esto además requiere el soporte del kernel y no es soportado muy bien. El servidor BootP deberá saber todo lo interno del IP asi como los detalles de la capa de Enlace de Datos.

Aunque el BootP es más poderoso que el RARP, si tiene algunas limitaciones. Este fué diseñado para un ambiente de redes estático y asi que los parámetros BootP de cada host son almacenados en un archivo en el servidor BootP. Cambios efectuados a la red requieren la edición del archivo de configuración del servidor. El BootP no puede manejar la asignación de parámetros dinámicamente. El DHCP si fué desarrollado para trabajar en ambientes dinámicos más modernos.

DHCP

El DHCP es una extensión del BootP. Su mejora principal son su gran número de parámetros de configuración que pueden ser pasados a los clientes, un método para que el cliente descubra información a cerca del servidor y la habilidad de asignar direcciones IP dinámicamente desde un pool. La versión actual del DHCP está especificada en los RFCs 2131 y el 2132.

Una convesación típica de un DHCP se describe en los siguientes pasos:

1. Primero una máquina “A” envia una trama de Descubrir-DHCP.
2. El servidor B (y puede ser otros servidores) responde con una Oferta-DHCP.
3. El Cliente “A” requiere una configuración al servidor “B” con una Petición-DHCP.
4. El servidor “B” envia la configuración con un Paquete-DHC.
5. El Cliente “A” ahora puede configurarse asi mismo.

Los pasos 3 y el 4 son los mismo como los de un BootP. Pasos 1 y el 2 son una extensión que permiten que el cliente encuentre investigue que servicios DHCP están disponible. En esta manera, el servidor DHCP puede servir a clientes BootP asi como a los clientes de un DHCP.

Reconectarse a Otra Red

Después de reiniciar, “A” trata de usar su configuración existente (DHCPRequest). “C” envia un DHCPNAck; esa configuración es inapropiada. Una sesión nueva DHCP se inicia. Ahora el cliente que fué previamente configurado vía el DHCP es conectada a una red diferente.

"A" envia su dirección IP anterior como parte del DHCPRequest. "B" puede entonces examinar esto (y otros detalles de configuración) para decidir si acepta la petición de "A" y le permite usar esa configuración.

Un proceso similar ocurre con la configuración dinámica cuando el alquiler de un cliente ha expirado. El cliente efectuá un DHCPRequest para ver si aún puede seguir usando su configuración existente.

Si un cliente desea desconectarse de la red y liberar su configuración, este puede enviar un DHCPRelease a su servidor. En esta manera, su dirección IP se incluye como disponible en el pool de direcciones sin asi tener que esperar que el alquiler expire.

Fijese que, al igual que con el BootP, existen identificadores de números enteros pasados en cada mensaje DHCP para que asi múltiples clientes y servidores puedan mantenerse informado el uno del otro.

Una limitación de DHCP es que no hay mecanismo alguno para interactuar con el DNS (Domain Name System), el cual mapea nombres de computadoras a direcciones IP. Las nuevas direcciones alquiladas no se agregan al DNS. Algunas implementaciones de DHCP pueden incluir cierto métodos de actualizar el DNS, pero en la actualidad no hay un método estándarizado para esto. Un nuevo sistema llamado DDNS (Dynamic DNS) está siendo desarrollado para sobre pasar estas limitaciones, agregar la información a la base de datos del DNS de manera que se le pasa al cliente. Como esta es la limitación principal, en el presente, del DHCP, se espera un rápido desarrollo en el futuro cercano.

Relay Agents

Ambos el BootP y el DHCP permiten al servidor residir en una red no local a traves del uso del concepto de un agente de relevo (relay agent). El relay agent pasa o reenvia las peticiones de los clientes desde una red al servidor y reenvia la respuesta también. No hay problema con confundir diferente clientes ya que el identificador de las direcciones de la Capa de Enlace es única.

Los enrutadores por lo general no reenvia o pasan paquetes de broadcast. Así que debe existir algún método que permita al servidor DHCP escuchar las peticiones desde otras subredes. Hay dos métodos para llevar esto acabo. Podemos instalar un DHCP relay agent (agente de relevo de DHCP) en cada subred que tiene clientes de DHCP o usar un enrutador que pueda ser configurado para pasar el broadcast en el servidor DHCP.

Otro método posible es configurar los enrutadores entre los segmentos. Lo configuramos para que reenvien los broadcasts al servidor DHCP/BootP al puerto UDP (puerto 67). Dependiendo en el enrutador, podemos tener el enrutador reenviar la petición directamente al servidor DHCP o generar otro broadcast.

Configurar un Servidor DHCP

La ISC (Internet Software Consortium) ha desarrollado una referencia de la implementaci´øn de cliente y servidor DHCP. La podemos encontrar en su página web http://www.isc.org/products/dhcp. En está sección cubriremos la configuración de un servidor DHCP.

Si usted compiló su kernel, será necesario que se asegure que incluyó el soporte de las operaciones multicast. La mayoría de los kernels que se incluyen en las distribuciones estándares traen este soporte habilitado. Podemos revisar si nuestro kernel lo tiene habilitado, observando la salida del comando ifconfig para observar si se incluye la palabra MULTICAST:

```
# ifconfig  eth0
eth0    Link encap:10Mbps   Ethernet    HWaddr  00:CO:1F:12:23:AB
inet addr:192.168.2.10    Bcast:192.168.2.255 Mask:255.255.255.0   BROADCAST   RUNNING  MULTICAST
MTU: 1500     Metric:!
```

Si tiene clientes de otros sistemas operativos aceptando servicios DHCP en la red, puede que deberá eje-

cutar el siguiente comando:

route add -host 255.255.255.255 dev eth0

Si es necesario entonces, tendrá que agregarlo a los scripts de inicio, quizas en el archivo /etc/init.d/rc.local.

dhcpd

El daemon servidor DCHP es llamado dhcpd. Por lo general es instalado en el directoio /usr/sbin, con un script de inicio en el directorio /etc/rc.d/init.d. Si su servidor DHCP es multihomed o actuá como un enrutador, tendrá que decirle al servidor DHCP en cual interface de red servir. Por ejemplo, si deseamos que sirva la red conectada a la primera tarjeta de red, deberá ejecutar la siguiente sentencia:

/usr/sbin/dhcpd eth0

Antes de iniciar el dhcpd, asegurece de que el archivo /etc/dhcpd.leases existe. Puede ser que diferentes distribuciones coloquen este archivo en otro directorio, como puede ser /var/state/dhcp/dhcpd.leases. usted puede lograr esto invocando el siguiente comando:

touch /etc/dhcpd.leases

Mientras pone a prueba el servidor DHCP, podemos ejecutar con debugging (modo de depuración de errores) habilitado y no ejecutarlo en el segundo plano o background, invocandolo de la siguiente manera y con las opciones aqui mostradas:

/usr/sbin/dhcpd -d -f

Ejecutelo en modo de debug, observando los mensajes al inicio de uno de los clientes de DHCP. Después de asegurarse de que los clientes están recibiendo si direcciones IP del servidor, agregue el daemon servidor a los scripts de inicio.

dhcpd.conf

El archivo de configuración principal del dhcpd es /etc/dhcpd.conf. Existen interfaces gráficas para editar este archivo de configuración, unas de ella es llamada kcmdhcpd, pero como el archivo es de texto plano tipo ASCII, lo más probable es que usará un editor de texto para configurarlo. La configuración tiene parámetros globales asi como secciones para establecer parámetros específicos a una subred o máquina.

Existen muchas opciones que pueden ser específicadas para ser enviadas al cliente y varias maneras de específicar que debe ser enviado a cada cliente. Aquí le presentamos un archivo dhcpd.conf de ejemplo:

```
default-lease-time 3600;
option subnet-mask 255.255.255.0;
option routers 192.168.2.1;
option domain name-servers 192.168.2.1, 192.168.3.1;
option domain-name “mi-dominio.org”;

subnet 192.168.2.1 netmask 255.255.255.0 {
    range 192.168.2.10 192.168.2.100;
    option netbios-name-servers 192.168.12.21;
}
host miservidor {
    hardware Ethernet 00:0C:12:43:AF:BE;
    fixed-address 192.168.2.13;
    option routers 192.168.2.1 192.168.2.2;
}
```

La primera línea establece el alquiler para los clientes a 3,600 segundos o 1 hora. La sentencia “option” específica la información que va a ser enviada a los clientes. Las primeras cuatro opciones son enviadas a los clientes DHCP.

La sección que inicia con “subnet” dice que cualquier cliente que este en el segmento 192.168.2.0 debe ser asignado una dirección IP a azar desde el pool de direcciones 192.168.2.10 al 192.168.2.100. Observe que la opción netmask es usada para especificar el tamaño de la subred que está definido encerrado entre las llaves, donde la opción subnet-mask específica la información que será enviada al cliente. La opción netbios-name-servers establece el servidor de nombre para los clientes de otro sistema operativo.

La sección que inicia con “host” muestra como establecer los parámetros para un host en específico. Usando este método, podemos especificar la dirección de hardware o MAC del sistema. recuerde que esto significa que si el NIC del sistema es cambiado, usted tendrá que reeditar el archivo de configuración del DHCP server. En este cso, específicamos que el sistema con la dirección MAC de valor 00:0C:12:43:AF:BE debe ser asignada la dirección IP número 192.168.2.13. La opción específicada en la sección host-specific anteceden en preferencia a las opciones en la sección global.

Configurar un Cliente DHCP

Básicamente todos los clientes DHCP efectuán la mismas tareas en diferentes formas. Ellos permiten que hosts obtengan una dirección IP dinámicamente cada vez que se efectuá un login o ingreso al sistema. Este protocolo es extremadamente importante en las redes amplias y complejas de hoy día. Usando el DHCP, múltiples hosts pueden ser administrados facilmente por el administrador atraves de una base de datos central sin tener que ejecutar un gran número de configuraciones a equipos individuales. El DHCP también permite que redes puedan acomodar un número de hosts mayor al número de direcciones IP disponibles, previendo, claro que no todos los hosts se conecten a la misma vez. Esta técnica es muy usada por los grandes ISPs que tienen una amplia matricula de subscritores con sólo un número limitado de direcciones IP que poseen para asignar. Ellos asumen que no todos los usuarios ingresarán al sistema simultáneamente. El DHCP también permite que un gran número de usarios moviles sean asignados una dirección y que se comuniquen sin importar que conexión física los une a la red, sin tener que cambiar su configuración.

Existen tres clientes principales para el protocolo DHCP: pump, dhcpcd y dhclient. Estos programas clientes interactuán con un servidor DHCP para obtener la información de configuración de la red. Ellos se ejecutan como daemons, siempre revisando para asegurarse que el alquiler aún es válido y renovarlo si es necesario. Si la conexión al servidor se detiene, entonces el programa cliente continuará tratando de contactar el servidor hasta que la conexión sea restaurada.

pump

El cliente pump es un paquete simple de DHCP desarrollado por Red Hat. Es el cliente más fácil de usar e incorpora las mismas características que otros paquetes mucho más difíciles de implementar. Se inicia automáticamente por el script /sbin/ifup pero puede ser ejecutado manualmente para revisar la configuración o el estado de la conexión. El comando pump tiene un número de opciones, pero la sintaxis de la línea de comandos es bien simple:

/sbin/pump -option

Esta es una lista de las opciones básicas:

-c	Define archivo diferente de configuración y no el /etc/pump.conf
-h	Nombre de host que se requiere.
-i	La interfaz a ser configurada (eth0, eth1, etc.)
-k	Mata/Kill el programa (el daemon y todas las interfaz)
-l	Tiempo de alquiler (horas)

-R	Renovación inmediata del alquiler al expirar
-s	Estado de la interfaz
-d	Detiene el cambio del DNS (sin cambios al /etc/resolv.conf)
-?	Help/Ayuda
--usage	Uso de la ayuda

Usted puede revisar el estado del daemon pump escribiendo:
/sbin/pump -i eth0

La salida sería algo parecido al siguiente ejemplo:
Device eth0
IP: 192.168.2.10
Netmask: 255.255.255.0
Broadcast: 192.168.2.255
Network: 192.168.2.0
Boot server 192.168.2.4
Next server 192.168.2.4
Gateway: 192.168.2.7
Domain: codigolibre.org
Nameservers: 192.168.5.54 192.168.5.254
Renewal time: Sun Jun 16 23:54:42 2004
Expiration time: Mon Jun 23 00:09:42 2004

Este comando retorna toda la información que fué asignada al cliente por el servidor DHCP. Aquí el usuario puede revisar cuando el alquiler expira y cuando el pump renovará el alquiler.

El pump mantiene la información de su configuración en el archivo /etc/pump.conf. Este es un archivo orientado a líneas que contiene comandos y argumentos parecidos a los comandos del shell, que incluye comillas y barras invertidas (\). Hay dos maneras diferentes de configurar un sistema: global o específica. La global afecta toda todas las conexiones manejadas por un daemon, mientras que la específica maneja conexiones únicas. Al definir el dispositivo con las opciones, los cambios pueden ser específicas a ese dispositivo; de cualquier otra manera, los cambios se efectuarán en todo el sistema. Aquí le presentamos un archivo de config de ejemplo que define un dominio para efectuar búsqueda, el número de reintentos a efectuar, y el dispositivo a configurar:

domainsearch “codigolibre.org abiertos.org”
retries 1
device eth2 {nodns}

Este archivo de configuración permite que cierto aspectos puedan ser cambiados con el comando pump desde la línea de comandos para ser integrados como constante en los arranques subsecuentes.

dhcpcd

El dhcpcd es un cliente DHCP poderoso que se puede usar con cualquier distribución de GNU/Linux. Este efectuá el mismo trabajo que el pump pero puede ser más difícil de usar. El dhcpcd tiene una línea de comadnos un poco más compleja debido al alto número de opciones necesarias para configurar una cosa, pump fué hecho para el usuario principiante, mientras que configurar con el dhcpcd es mejor dejarselo a los usuarios experimentados.

La línea de comando, dhcpcd [-opción argumento] [interfaz], se ve simple, pero entrar en la opciones toma cierto tiempo. Aquí es una pequeña lista de opciones disponibles y sus próposito:

d-d	Escribe al log syslog con mucho detalle.
-k	Mata/Kills la conexión actual y elimina el cache (permite nuevas peticiones de direcciones).
-B	Hace una petición de una respuesta tipo broadcast desde el servidor.
-D	Obliga al dhcpcd a usar el nombre de dominio retornado por el servidor.
-H	Obliga al dhcpcd usar el nombre del host retornado por el servidor.
-R	No cambiar el DNS; no reemplazará el archivo /etc/resolv.conf.
-t	Timeout/Tiempo agotado (en segundos).
-c filename	Ejecute el archivo de nombre filename (por lo general un script).
-h hostname	Específica un nombre de host que le enviará al servidor (como un login).
-i VTD	ID Vendor.
-I CID	ID Cliente.
-l	Envia el tiempo de alquiler recomendado (en segundos).
-s IP	Hace la petición de una dirección IP específica; si se deja en blanco, devolverá la IP actual.
-s IP interface	Define la interfaz a ser configurada.

El dhcpcd utiliza múltiple archivos para contener la información que ha capturado. La mayoría de estos están localizados en el directorio /etc/dhcpcd. La parte <interface> de los archivos es reemplazada con el dispositivo que usted está usando. Se produce un archivo por cada dispositivo que va a ser configurado. Aquí en lo adelante se listan los archivos necesarios y el propósito de cada uno.

/etc/dhcpcd/dhcpcd-<interface>.info
Este contiene la información que el dhcpcd ha adquirido; es usado para mantener los mismos parámetros cada vez si e posible.

/etc/dhcpcd/dhcpcd-<interface>.exe
Este se ejecuta cada vez que es cambiado el IP del sistema; este actualiza toda la información del sistema.

/etc/dhcpcd/dhcpcd-<interface>.pid
Este contiene la ID del proceso requerido por el servidor.

/etc/dhcpcd/dhcpcd-<interface>.cache
Este contiene la dirección previamente asignada usada por el dhcpcd para readquirir la misma dirección en cada conexión; esta puede ser limpiada co la opción -k.

/etc/resolv.conf
Este archivo contiene las opciones de nombre de dominio y DNS retornada por el servidor; es limpiado cada vez que una nueva conexión es efectuada al menos que la opción -R es usada.

Es importante señalar que no debe confundir el daemon servidor, dhcpd, y el daemon cliente, dbcpcd.

dhclient

El dhclient es un cliente DHCP mucho más simple para las distribuciones que no soportan pump. El dhclient utiliza un archivo de configuración simple en vez de de cambiar la configuración desde la línea de comandos. la línea de comandos es dhclient -option con -p port para especificar el puerto que el dhclient transmite y -d para ejecutar el dhclient en modo de depuración o debug mode.

El dhclient revisa el archivo /etc/dhclient.conf antes de inicializar las instrucciones de configuración. El archivo contiene toda la información necesaria por el dhclient y la información adquirida dsde el servidor DHCP. Aquí le presentamos un ejemplo del archivo dhclient.conf:

```
$ dhclient.conf
timeout   60;
```

```
retry 60;
reboot  10;
select-timeout  5;
initial-interval   2;
interface  “eth0”   {
      request  subnet-mask,   broadcast-address, time-offset,
      routers,  dhcp-lease-time,
      domain-name,  domain-name-servers, host-name;
      require subnet-mask,  domain-name-servers;
      send dhcp-lease-time  24000 ;
}
```

El archivo es similar al archivo de configuración de pump. Podemos específicar opciones globales o específicas, como se demuestra aquí con la opción interface. Fijese, como la llaves {} deben ser usadas para asignar opciones asociadas con una interfaz, el dhclient también necesita un archivo dhclient.leases. Este archivo contiene toda la información de todas las direcciones IP que han sido asignadas, el dhclient agregará nuevas entradas al final del archivo. Pero estas adiciones pueden causar un problema. Listas masivas de alquiler que consisten en miles de viejos alquileres toman mucho espacio de disco, el dhclient resuelve esta problematica crando un nuevo archivo de alquiler periodicamente, dejando los alquileres anteriores en archivos por separados de resguardo/backup. Estos archivos de backup pueden entonces ser eliminados si es necesario o retenidos para cuestion de record.

Ejercicios 7-2: Configurar el DHCP

En este ejercicio se rquieren dos computadoras conectadas en red. Instalaremos un servidor DHCP en una de ellas. Necesitaremos un cliente en la otra para hacer la operación de prueba. En este ejercicio deberá usar la dirección de subred privada 192.168.0.0. No se proveen soluciones para este ejercicio.

1. Instale el dhcpd. En muchas distribuciones, ya este estará instalado. Si no es el caso, es recomendable que usted instale el paquete proveido por su vendor. Alternativamente puedes descargar el fuente desde el ftp.isc.org descomprimirlo, desempaquetarlo, compilarlo e nstalarlo.

2. Determine donde se encuentra su archivo dhcpd.leases. Puede que este en /etc o quizas en /var, como en /var/state/dhcp. Las páginas man de su paquete le puede ayudar a ubicar su archivo dhcpd.leases:
```
# man 5 dhcpd.leases
```

3. Si el archivo dhcpd.leases no existe deberá crearlo:
```
# touch  /etc/dhcpd.lease
```

4. Edite el archivo /etc/dhcpd.conf para que contengan las siguientes líneas:
```
dcfault-lease-time 3600;
option  subnet-mask  255.255.255.0;
option  routers  192.168.0.1;
option domain-name-servers  192.168.0.1,   192.168.0.9;
option domain-name  “mi-dominio.org”;

subnet  192.168.0.0  netmask  255.255.255.0 {
      range  192.168.1.20   192.168.1.250;
      option  netbios-name-servers   192.168.0.15;
}
```

5. Inicie el Servidor DHCP en modo de depuración (debug):
```
# /usr/sbin/dhcpd   -d   -f
```

6. Inicie el cliente DHCP. En el sistema que debe estar configurado para usar DHCP y luego podemos revi-

sar para ver que parámetros nos asigno.

7. Observe la salida de depuración que el daemon servidor dhcpd expide.

8. Revise los parámetros que el cliente DHCP recibió. En GNU/Linux, con en la mayoría de sistemas operativos podemos usar los comandos ifconfig y route sin argumento alguno.

9. Renove el alquiler de la dirección que recibió. Con el cliente de pump, usted puede efectuar esta tarea escribiendo la siguiente sentencia:
pump --renew

10. Verifique si puede renovar su alquiler de DHCP y recibir uno nuevo. Con el cliente pump, usted puede lograr esta tarea primero relegando el alquiler y luego contratar uno nuevo:
pump --release
pump --renew

Con el dhcpcd, tendrá que matar el daemon, deshabilitar la interfaz y rehabilitarla.

TELNET

El servicio de Telnet es uno de lo más usado en los sitemas UNiX y lo más seguro es que usted ya este familiarizado con este. Telnet provee acceso a una conexión de terminal virtual en una red. Esta conexión virtual posibilita ejecutar programas de consola igual que si estubiese sentado en el equipo remoto. La naturaleza del uso de la línea de comandos de GNU/Linux y UNiX apodera al administrador de sistema manejar su sistema de cualquier ordenador y de cualuier punto en la red.

El servidor telnet escucha por conexiones entrantes en el puerto TCP 23. El servidor acepta conexiones desde los clientes, luego inicia un proceso de login a esa conexión. En GNU/Linux, el servidor utiliza una técnica conocida como “pseudo terminal” para simular un terminal físico. Los pseudo terminales son dispositivos de software que simulan el comportamiento de un terminal, aunque no esten conectadas a un dispositivo físico sino a algún tipo enlace de comunicación. Ellas forman la base de la gran mayoría del software de emulación remota en GNU/Linux. Hasta en los emuladores de terminales locales como es el xterm se emplea el uso de pseudo terminales. Un pseudo terminal es mejor conocido como un pty.

La comunicación de los caracteres entre el cliente y el servidor se lleva acabo utilizando un formato independiente de máquina conocido como NVT (Network Virtual Terminal). Es responsabilidad del cliente y el servidor en cada punta convertir entre el formato NVT y las caracteriísticas del teminal del sistema local.

El telnet es asignado un único canal para los comandos y la data por definición del RFC 854. Telnet permite que los clientes inicien y tomen parte de sesiones interactivas en sistemas remotos. El equipo cliente es capaz de iniciar sesión en el servidor para asi poder transferir data y llevar acabo otras actividades interactivas.

El servidor daemon de telnet es por lo general ejecutadod esde inted, asi que usted puede usar los wrappers de TCP para filtar acceso. Su ejecutable es nombrado in.telnetd o telnetd. The following represents the entry chat you should use in /etc/inetd.conf using TCP wrappers:
telnet stream tcp nowait root /usr/bin/tcpd in.telnetd

Una característica interesante del programa telnet es que usted puede ejecutar un telnet a cualquier puerto TCP. Cuando hacemos esto, usted termina comunicandose con el servicio de red y por ende debe usar los comandos raw definidos por el servicio. De hecho, no tendrá ningún software cliente y deberá actuar como

el programa cliente mismo. En el siguiente ejemplo ilustramos una conexión a un sistema FTP directamente:

```
$ telnet ftp.abiertos.org 21
Trying 192.168.2.15...
Connected to ftp.abiertos.org.
Escape character is '^]'.
220 ftp.abieros.org FTP server (Version 2.6.0) ready,
user anonymous
331 Guest login ok. send your complete e-mail address as password.
pass mike@abiertos.org
230-Welcome to My FTP Site.
230 Guest login ok. access restrictions apply.
cwd /pub
250 CWD command successful .
pwd
257 "/pub" is current directory.
type i
200 Type set to I.
size archivo.txt
213 1416
quit
221 Goodbye.
```

En esta sección, discutiremos tópicos que concernientes de seguridad.

Seguridad y Telnet

Telnet no debe ser usada en conexiones abiertas atraves del Internet. Cualquier contrasenna que escriba puede ser interceptada ya que telnet no las encripta. Este las envia por la red en formato de texto plano. Para el acceso seguro de un shell, deberá usar el programa SSH (Secure Shell).

El SSH provee encriptación entre el cliente y el servidor. Este método de transferencia encriptada protege tanto las contraseñas como la información que se transmite entre los dos sistemas. Sin tomar en cuenta esta seguridad agregada, el SSH funciona igual que el telnet.

Hasta en un ambiente corporativo, usted debiera considerar utilizar los servicios del SSH. Como el Ethernet es un medio compartido, cualquier sistema en cualquier segmento que la data atraviesa puede escuchar a todo que pasa atraves de el. El Token Ring es también un medio compartido. Nunca se puede estar seguro de quien está huzmeando nuestro tráfico. Recuerde que un gran porcentaje de los ataques a los sistemas son cometidos desde adentro. Utilizar switches en su red agrega un nivel mayor de de seguridad ya que solamente direcciona paquetes al destino final. Aún asi, la data deberá viajar atraves de enrutadores y hasta redes desconocidas.

SSH/OPENSSH

Es obvio la necesidad de inclementar los nivels de seguridad entre las redes a medida que que más y más personas utilizan el Internet para intercomunicarse. En el pasado, los métodos de comunicación con aplicaciones como es el telnet no ofrecen niveles aceptables de seguridad. Cuando un usuario se conecta remotamente, existe el potencial de ser monitoreado por una fuente externa. Esta fuente externa puede ser un cracker o cualquiera intentado ingresar ilegalmente a nuestra red. La seguridad para los acceso remotos debe ser aplicado igualmente a los usuarios del hogar y los corporativos en su uso del Internet. Para contrarestar la violaciones de la red, un método ha sido desarrollado para proveer niveles de seguridad más alto para los ingresos interactivos remotos llamado Secure Shell (SSH).

Existen dos versiones del SSH. La versión 1 fué lanzada de licenciamiento libre y es mantenida como el

OpenSSH. La versión 2 opera bajo protocolo diferente y es sólo libre para su uso no comercial. Desde el punto de viasta del usuario, ambas operan de la misma manera y con los mismos resultados. El comando ssh lanza ambas versiones.

El usuario utilizza el programa de generar llaves ssh-keygen para generarse un par de llaves privada y pública, luego coloca la llave pública en el host remoto en un de dos directorios $HOME/.ssh/authorized_keys para el SSHl o el directorio $HOME/.ssh2/authorization para el SSH2. Una vez la llave pública ha sido colocada en el host remoto, los usuarios locales pueden ingresar remotamente al host usando la clave (passphrase) secreta del SSH.

Los siguientes tópicos son discutidos en esta sección:

- SSHl
- SSH2
- OpenSSH y Stunnel
- Instalación
- Configuración
- Usado

SSH1

El SSH1 es la primera versión del programa Secure Shell y es usada para ingresar remotamente a un sistema. Es la manera segura de obtener una sesión de login interactiva segura. La función principal del SSH1 es la de encriptar cada paquete que atraviesa desde una máquina hacia una red externa. Esto previene que los crackers puedan decodificar los paquetes después de interceptarlos. Si un cracker logra interceptar nuestros paquetes, la desencriptación de nuestro paquete será una ardua tarea. Asi que, el chance de que nuestra información sea explotada exitosamente son muy pocos. En resumen, usar el SSH al conectarse remotamente a una red protejerá su información mucho más seguro que usando telnet y cualquier otro cliente. El programa SSH1 soporta cuatro diferente tipos de cifrados para su encriptación: DES (Data Encryption Standard), 3DES (Triple DES), IDEA (International Data Encryption Algorithm), y Blowfish. El cifrado soportado para autentificación es RSA, que son las consonantes de los nombres Rivest, Shamir y Adleman del cifrado de encriptación. El SSH1 ya no es desarrollado en la actualidad.

SSH2

La nueva versión del programa Secure Shell es el SSH2, el cual ofrece una mejor estructura para el protocolo. La implementación básica del SSH2 es completamente diferente a la del SSH1. Como mencionamos anteriormente, el SSH encripta los paquetes de data antes de enviarlo y los desencripta en el host remoto. El SSHl y SSH2 son distintos en que ellos encriptan la data en doferentes puntos del paquete, haciendo la encriptación diferente en estructura y composición. El SSH2 posee una encriptación más altaque provee mejor seguridad, mejor y más confiable rendimiento y esta disponible en varias ediciones y para una variedad de sistemas operativos. Estas nuevas características hacen que el SSH2 sea un estándar de seguridad deseado. El SSH2 tiene 5 cifrados para encriptar: 3DES, Blowfish, Twofish, Arcfour y Cast 128-cbc. El cifrado soportado para la autentificación es el RSA (Digital Signature Algorithm).

OpenSSH y Stunnel

OpenSSH es un paquete de programas que contienen un conjunto de herramientas de red útil para la implementación del protocolo SSH. Entre estas herramientas se incluyen el programa SSH y los comandos sshd, ssh-agent,ssli-add,scp y ssh-keygen. Estos comandos serán brevemente descripta más adelante:

sshd

El Secure Shell Daemon se ejecuta constantemente y escucha por otros hosts que intentan establecer una conexión segu-

ra con SSH.

ssh-agent

Es activado cuando uno ingresa a una localidad remota. Este proteje la llave RSA usada en la autentificación.

ssh-add

Este comando agrega y actualiza las identidades al agente de autentificación.

scp

El coamndo scp oermite la copia segura de archivos desde un host en una red a otro.

ssh-keygen

Este genera una llave para que los paquetes tanto entrante como saliente tengan autentificación y sean aceptados y aprovados.

El programa stunnel es similar al SSH en que este encripta los paquetes para una comunicación segura. El stunnel utiliza autentificación SSl (Secure Socket Layer) con las librerías OpenSSL. Una ventaja de este programa ocurre cuando un sistema se encuentra ejecutando un daemon SSL y la computadora que se comunica no está ejecutando SSL. Un canal de comunicación segura puede ser establecida sin importar el conflicto. El stunnel también puede protejer contra la intercepción y la manipulación de data durante la transmisión a su destino primario.

Instalación

OpenSSH se incluye con la mayoría de las distribuciones y el fuente puede ser obtenido desde el portal http://www.openssh.com. SSH2 puede ser encontrado en http://www.ssh.com. El SSH original es soportado al igual que el protocolo SSH2 en las versiones 2.3 en adelante del OpenSSH. Después de extraer el archivo .tar, el fuente deberá ser configurado con el comando ./configure. Luego, deberá ser compilado con el comando make. Finalmente, tendrá que instalar el software con el comando make install.

Configuración

El servidor lee su configuración desde el archivo /etc/sshd_config y el cliente lo lee desde /etc/ssh_config y entonces desde uno $HOME/.ssh/ssh_config para el SSH o desde el $HOME/.ssh2/ssh2_config para el SSH2. Algunos administradores de sistemas puede que deseen aliviar algunas de las restricciones por defecto editando el archivo sshd_config.

Uso

El comando ssh debe ser ejecutado por lo menos con un argumento- el nombre del host al cual deseamos conectarnos. Si ningún otro argumento es suplido, un shell en el host remoto se iniciará, ssh también puede ser ejecutada en el formato:

ssh HOST COMANDO

el cual puede causar que ssh ejecute el COMANDO en el HOST remoto. Si el comando a ser ejecutado es interactivo (por ejemplo, un cliente de ftp o correo), la opción -t debe ser usada (ssh -t HOST COMANDO), o (con el SSH2 solamente) use el la directriz ForcePTTYAllocation en el archivo sshd_config.

El ssh también puede reenviar conexiones X11. Esto permite que un usuario se conecte a programas desde el host remoto a su display X local y con un alto nivel de seguridad a diferencia del VNC. Este debe ser habilitado con la directiva ForwardX11 en el archivo sshd_config.

El comando para ingresar a un sistema que tiene ssh instaldo y ejecutándose es parecido al siguiente:

$ ssh [-l login name] hostname [command]

El parámetro -1 permite al usuario a especifícar un nombre para el login. El nombre de host es un parámetro requerido para representar el computador que se desea establecer la conexión. Una lista de opciones más completa se encuentra en la página man del comando SSH.

La aplicación de ssh debe usarse cuando se desea conectar a una localidad remota. Por ejemplo, para

conectarse a codigolibre.org para revisar el espacio en disco disponible, la sentencia ssh sería así:

$ ssh -l jazmine codigolibre.org

En este comando ssh, se muestra como ingresar al sistema codigolibre.org usando el nombre de usuario jazmine. La opción -1 pasa el nombre del usuario para esta sesión, jazmine, al programa. En algunos sistemas, los argumentos se pueden presentar en al reves, asi que esta sentencia también es permitida:

$ ssh codigolibre.org -l jazmine

Despué que una conexión segura ha sido establecida al host, el servidor le pedirá la contraseña para la cuenta del usuario sometido después de la opción -l.

Al usar una llave pública de autentificación, el usuario puede usar ssh-agent para sólo abrir la llave privada una vez, el ssh-agent es mejor usado para iniciar el shell del usuario o una sesión del X. El administrador puede establecerlo finalizando su script del sistema del arranque del X con la sentencia:

exec ssh-agent $HOME/.xsession

o el usuario puede habilitarla finalizando su archivo $HOME/.xsession con:

exec ssh-agent WINDOW-MANAGER

Para que el ssh-agent pueda hacer cualquier cosa, las llaves deberán ser agregadas con el comando ssh-add. Al ejecutar, ssh-add le pedirá la clave o passphrase para desbloquear la llave privadato . Asi que, todos los procesos SSH que son decendientes del proceso ssh-agent process no le pedirán la clave o passphrases ya que la autenficación se ha llevado acabo.

El SSH biene además con el programa lamado scp (Secure Copy). Es usado para hacer copias seguras de archivos desde un host a otro. Aunque es un reemplazo para el ftp, SSH2 también incluye un reemplazo aparte del ftp llamado sftp. El scp es ejecutado así:

scp Fuente Destino

Donde Fuente es en el fomato de host:nombre-archivo. También se pueden suplir los nombres de usuarios, por ejemplo:

scp archivo root@codigolibre.org:/etc

El sftp es similar al ftp y soporta los mismo comandos básicos del ftp como lo son: ls, cd, put, get, etc.

FTP

El FTP es el Protocolo de Transferencia de Archivos (File Transfer Protocol), que provee la transferencia de archivos de texto y binarios. Es uno de los protocolos IP básicos y es casi tan antiguo como el propio TCP/IP. En esta sección, daremos un vistaso a la parte básica del protocolo FTP y el uso básico de tanto el cliente como el servidor FTP. Además cubriremos las implementaciones particular de los servidores wu-ftpd y ProFTPD.

Aunque muchas transferencias de archivos son efectuadas usando HTTP, el FTP continua presentando algunas ventajas. Como por ejemplo, es mucho más fácil resumir el transfer si se detiene y el FTP también mantien los bits de información de los archivos como son los permisos y las fechas de modificaciones. El FTP permite al cliente navegar atraves de un sistema de archivos en el servidor, proveiendo un sistema de archivos virtual que no está disponible en el HTTP. El FTP también puede transferiri múltiples archivos en una sóla operación.

Los siguientes tópico son discutido en esta sección:

• Clientes FTP
• Servidores FTP
• El Protocolo FTP

• wu-ftpd
• ProFTPD

Clientes FTP

El usuario final accesa los servidores FTP desde un programa cliente. El cliente FTP más usado es el programa que se ejecuta desde la línea de comandos. Este inicia una conexión con un servidor, luego usted escribe comandos para recibir información y entonces da inicio a la transferencia de los archivos. El cliente traduce los comandos que usted escribe a comandos del protocolo FTP y se los envia al servidor. Este entonces interpreta las respuestas del servidor y se los presenta al usuario.

Otro cliente muy utilizdo en GNU/Linux es el cliente de FTP del navegador Mozilla. Aunque su uso principal es navegar en páginas de servidores HTTP, pero Mozilla también tiene pacidad de interactuar con servidores FTP. Si dirige al browser hacia una dirección URL del tipo ftp (e.j., ftp.codigolibre.org) o das click sobre un enlace a una URL de un FTP, Mozilla enviarea los comandos apropiados al servidor FTP y desplegará el listado de los directorios o descargará el archivo. Fijese de que manera podemos incluir el nombre de usuario y contraseña dentro del URL, algo similar a esto:

ftp://nombreusuario:contraseña@ftp.redhat.com/

Tome en cuenta que su contraseña será transferida en texto plano, asi es que tenga cuidado al usar esta manera de conectarse a un servidor.

Muchas distribuciones proveen clientes FTP que proveen interfaces gráficas y que son muy intuitivos, entre ellos se encuentran: gftp y NcFTP. Ellos proveen mecanismos de almacenar bookmarks a sitios muy visitados por el usuario y barras de progresos de las descarga, entre otras cosas para asistir la navegación. Estos también simplifican los comandos estándares de los clientes de línea de comandos y agregan características como son: completar nombres, descargar múltiples archivos, descarga en segundo planno, auto resumir descargar si falla, entre otras. Si usted deberá continuamente usar un cliente FTP, entonces lo más recomendable es usar uno de estos clientes.

Servidores FTP

Los servidores FTP permiten que otros sistemas remotos accesen a una porción de sus sistema de archivos. El FTP proveela misma funcionalidad del HTTP pero más orientada a la transferencia y no a la presentación de los archivos. El FTP es mucho mejor para la transferencia de archivos binarios y es muy frecuentemente usado para la descarga de software.

Un servidor FTP permite que solo usuarios autorizados usando un mecanismo estándar de nombre/contraseña puedan iniciar una sesión, o puede ser configurado para permitir el acceso a usuarios anonimos (guest users). Lo de usuario anonimo son referidos como anonymous FTP server y es la manera primordial de la transferencia de software Libre. El acceso a FTP Anonimo agrega una problematica de seguridad. Por ejemplo, si deseamos que los usuarios no puedan accesar archivos del sistema de archivos como es /etc/passwd. En la mayoría de los casos, usted tampo deseará que ellos puedan subir o escribir archivos al servidor ya que usted no controla quien se conecta por la cuenta anonimo.

Los servidores FTP muy a menudo utilizan la llamada del sistema chroot para mejorar los niveles de seguridad. La llamada chrootcambia el directorio raíz de un proceso. Después de la llamada chroot, los procesos no pueen aaceder ningún archivo fuera del subdirectorio que fué especificado en la llamada. El proceso nunca puede deshacer la llamada chroot y los procesos hijos heredarán el ambiente chrooted. El uso del chroot en los servicios FTP le permiten asegurarse que los usuarios anonimos sólo tendrán acceso limitado a un subconjunto de directorios en su sistema.

El Protocolo FTP

El protocolo FTP fué descrito por primera vez en el RFC 172 en el año 1971, aún antes de la implementación actual del TCP/IP fuese desarrollada. El protocolo actual está descrita en el RFC 959. Varias clarificaciones y actualizaciones se le han efectuado, incluyendo RFCs 1123, 1579, 2228, 2389, y el 2428.

Por RFC 854 se le asignan dos canales al FTP. Un canal es para los comandos y respuestas, y el otro es para la data y los archivos.

Una conexión FTP realmente consiste en dos conexiones, una para los comandos y otra para la data. La conexión de control se mantiene siempre abierta, pero la conexión de data sólo se abre cuando es requerida. La conexión de data del lado del servidor se encuentra en el puerto 20 y la de control se encuentra en el puerto 21. En realidad la conexión de data es abierta por el servidor, no el cliente. El cliente se enlaza con un puerto efímero y envia ese número al servidor. El servidor luego toma el puerto 20 de su lado y establece la conexión entre ese puerto y el puerto efímero del lado del cliente. La data del archivo entonces será pasada por esta conexión, con el fin de archivo definido por el que envia la data cuando ciuerra la conexión.

Básicamente, la manera que un FTP funciona es que el cliente inicia una conexión con un servidor. Este luego le envia comandos sobre esa conexión para recibir información y se inicia la transferencia de data. Por cada comando enviado, el servidor responde con resultados númericos además de una descripción en texto como resultado o cualquier error. La transferencia de data ocurre vía una segunda conexión. Al momento que una transferencia de data va a ocurrir, el clieente y el servidor negocian para establecer la conexión de data.

Existen dos modos deestablecer una conexión de data. En el estándar modo PORT, el cliente escucha en un puerto TCP y el servidor se conecta a el. Pero esto puede causar problemas con los firewalls, ya que las mayoría de cortafuegos no permiten conexiones entrantes. Asi que las nuevas versiones del protocolo FTP implementan modo pasivo. En este modo, el servidor escucha en un puerto por separado y el cliente efectuá una segunda conexión al servidor. La única diferencia entre los dos modos es si el cliente o el servidor inician el canal de data.

El FTP es un protocolo basado en comandos en que comandos ASCII son pasados entre el cliente y el servidor. Cuando un cliente FTP hace una conexión, este envía un comando ASCII al servidor. Estos comandos son quizas un poco primitivos y por lo general son mapeados a una sintaxis un poco más amistosa por la interfaz del cliente FTP. El servidor responde con un código numérico de tres dígitos y un mensaje opcional para indicar éxito, error, información de ayuda o la necesidad de más información.

Los que a continuación les presentamos son algunos de los comandos que son usados en el protocolo FTP:

USER	Específica el nombre del usuario del login.
PASS	Específica el password/contraseña del login.
CWD	Cambia el directorio de trabajo.
LIST	Lista los archivos en el directorio.
TYPE	Específica el modo de transferencia de binario o texto.
RETR	Descarga (Retrives) un archivo.
STOR	Almacena (sube) un archivo.
REST	Reinicia una transferencia incompleta.
DELE	Elimina (Deletes) un archivo.
QUIT	Cierra la conexión.

Recuerde que por lo general no tendrá la necesidad de utilizar estos comandos ya que el cliente le presentará un conjunto de comandos diferentes para efectuar las mismas acciones. Pero, si usted se conecta por telnet al servidor FTP en el puerto 21, usted puede usar estos comandos para ingresar al sistema y obtener cier-

ta información. Usted puede efectuar comandos raw y ver las respuestas raw, con los códigos de error y mensajes de texto. Usted puede ver un ejemplo de esta canal de comunicación raw en la sección anterior de telnet. Usted no podrá descargar archivos usando este m´todo porque el protocolo FTPrequiere el establecimiento de una segunda conexión para la transferencia de archivos.

wu-ftpd

Uno de los servidores FTP más usados es el wu-ftpd. Está disponible desde su sitio FTP propió, ftp.wu-ftpd.org. La mayoría de distribuciones de GNU/Linux vienen con wu-ftpd pre-instalado y a menudo es instalado con el nombre in.ftpd.

El servidor wu-ftpd provee muchas extensiones sobre el antiguo servidor basado en BSD FTP. Este provee capacidad de logging/diario, compresión dinámica (on-the-fly) y la capacidad de archivar (tar archiving), permisos por directorios de subir archivos, mensajes de login y directorio, limites de login basado en la clasificación de usuarios y hosting virtual.

La configuración del wu-ftpd se lleva acabo en varios archivos. Todos se encuentran en el directorio /etc por defecto. El archivo de configuración más importante es ftpaccess. Este define clases de usuarios y cualquier limitación colocada en esas clases. Algunos de los campos que pueden ser configurados en el archivos ftpaccess son:

- Dirección e-mail del Administrador
- Clases de Usuarios
- Limitaciones en el número de conexiones
- El archivo del mensaje de vienvenida
- Compresión y tar para todos los usuarios
- Diario/Log de las Transferencias
- Restricciones en las transferencias de archivos
- Restricciones en los comandos para los usuarios anonimos
- Habilitar y deshabilitar contraseña para los usuarios anonimos

Para habilitar el uso del archivo ftpaccess, deberá iniciar el daemon con la opción -a.

El primer pareametro que quizás desee establecer es las clases de los usuarios. Las clases de los usuarios dividen a los usuarios que se conectan al servidor para que asi le pueda asignar diferente ancho de banda y limitar su número de ingresos al sistema. Puede efectuar algo como este ejemplo a continuación:

```
class local real.anonymous 192.168.12.* * .codigolibre.org
class remote anonymous *
limit remote 10 Any /etc/ftp/demasiado.txt
```

El parámetro limit establece cuantos usuarios de cada clase pueden logearse en un espacio de tiempo y un mensaje a desplegarle si se le deniega el acceso. Usted puede permitir que suban archivos a directorios especificos utilizando los parámetros de subida. En este, usted puede especificar el directorio raíz, el directorio desde el punto de vista del usuario, si se puede subir archivos a este directorio, el dueño y el grupo que son los propietarios de los archivos y los permisos que deben recibir los archicos:

```
upload class=local /home/ftp /incoming yes owner group 0644
```

El control sobre la compresión on-the-fly y archivar es administrada por el archivo ftpconversions. De hecho, en el usted específica la extensión del archivo que existe, la extensión que el usuario está requieriendo y el comando para convertir entre los dos, acompañado de algunas opciones. Para habilitar las características de tar y de comprimir, necesitará especificar las opciones de tar y compress en el archivo ftpaccess.

El archivo ftpusers específica los usuarios que existen en el archivo /etc/passwd que no son permitidos usar el servicio FTP. De cualquier otra forma, cualquier otro usuario en el archivo /etc/passwd está permitido usar el servicio de FTP siempre y cuando el shell especificado en el archivo passwd está listado en /etc/shells. Aquí le presentamos una parte de un archivo ftpusers:

```
news
nobody
root
```

Podemos ejecutar el wu-ftpd como un servidor stand-alone si le especificamos la opción -S. Al ejecutarlo stand-alone, usted puede especificarle el puerto en el que escuche con la opción -p. Por lo general, usted puede ejecutar wu-ftpd desde el inetd, con una línea parecida a la que sigue en el archivo inetd.conf:

ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -a -l

No se olvide de enviar una señal HUP al inetd si usted efecuá cualquier cambio a su archivo de configuración inetd.conf. Fijese que usamos unwrapper TCP sólo para ser consistente; la opción -a habilita el control de acceso basado en host usando el archivo ftpaccess. La opción -l nos indica que deseamos habilitar escritura al log.

ProFTPD

Un popular servidor FTP de licencia GPL es el ProFTPD, disponible en su sitio web para descarga http://www.proftpd.net. Escrito desde cero con la intensión y meta de ser seguro y con más características que cualquier otro servidor FTP. Este usa un archivo central de configuración, con un grupo de directivas que deben ser muy familiar a todo administrador que está familiarizado con el Servidor Web Apache. El ProFTPD soporta logging, hosting virtual, autentificación PAM, integración SQL, integración con LDAP, mensajes de login y directorios entre muchas otras características.

La configuración .ftpaccess es similar a la que encontramos en el Apache en el archivo .htaccess. Estos archivos contienen los nombres de usuarios y su contraseña encriptada y son usados para restringir acceso a los usuarios a los directorios y archivos.

El Servidor ProFTPD puede ser ejecutado como un servicio stand-alone o desde el super daemon inetd. Para ejecutarlo desde el inetd, elimine cualquier otra línea que se refiera a otro FTP y reemplacela con la siguiente:

ftp stream tcp nowait root /usr/sbin/tcpd in.proftpd

Recuerde que deberá enviarle la señal al inetd de leer nuevamente si archivos de configuración:

killall -HUP inetd

Configuración

Uno de los componentes más útiles del ProFTPD es la directiva AllowFilter. AllowFilter puede establecer una expresión regular que debe ser igualada para cualquier comando que sea enviado al ProFTPD. Esto le permite al administrador controlar con precisión cuales caracteres puede ser enviado al ProFTPD. El AllowFilter depende en igualar el argumento a una expresión regular, asi que el administrador debe ser muy cuidaso al crear la expresión. Si un comando falla la prueba de la expresión, el sistema retornará el mensaje de error “Forbidden Command” a la persona intgresada al sistema. Estos filtros son colocados en el archivo de configuración del servidor. Un ejemplo de una expresión de permitir (allow) que sólo aceptará caracteres alfanumericos y espacios es:

AllowFilter “.*/[a-zA-Z0-9]+$”

Esta entrada establece una expresión que busca caracteres que igualan cualquier de estos caracteres ., *, /, +, $, o cualquier letra mayuscula o minusculas y cualquier número.

Similar a la directiva AllowFilter, la directiva DenyFilter utiliza una expresión regular a igualar. La diferencia es que en el caso de DenyFilter el coamndo debe NO igualar la expresión. Si el comando enviado al servidor iguala la expresión de denegar, el usuario ingresado en el sistema recibirá el mensaje de error "Forbidden Command" ("Comando no Permitido"). Estos filtros también son establecidos en el archivo de configuración. Aquí le presentamos un ejemplo:

DenyFilter "%?"

Este denegará cualquier comando enviado al servidor que contengan uno de los caracteres % o ?. Otra directiva es CommandBufferSize, la cual permite al administrador controlar el tamaño, een longitud de caracteres, que el servidor aceptará.

En resumen, el ProFTPD es un servidor FTP poderoso, que está disponible para GNU/Linux y todos los sistemas operativos Tipo-UNiX. Bajo constante desarrollo, con nuevas características y mejoras siempre disponible. Si un sistema esta ejecutando una versión que no es la actual, debe ser actualizada, porque como digimos ProFTPD está siempre mejorando. En particular, cualquier versión anterior a la 1.2.9 tiene cierto riesgos de seguridad que en las versiones superiores han sido todas corregidas. Todo sistema por seguro que sea tiene sus fallas asi es que porque ProFTPD es muy seguro no se descuide. Dos mejoras aún por materializarse son que no soporta contraseñas encriptadas y ID seguras, si su compañía requiere niveles de seguridad más estrictos, les acosajamos usar el VsFTPD. El cual si incluye las carecterísticas faltante de ProFTPD.

El servidor ProFTPD almacena su configuración en el archivo /etc/proftpd.conf. Aquí le presentamos un configuración de ejemplo. Discutiremos sólo los pareametros más comunes.

```
ServerName              "FTP de CodigoLibre"
ServerType              inetd
DefaultServer           on
Port                    21
MaxInstances            30
User                    nobody
Group                   nobody
Umask                   022
<Directory  /*>
    AllowOverwrite      no
</Directory>
<Anonymous   /home/ftp>
User                    ftp
Group                   ftp
UserAlias               anonymous ftp
MaxClients              10
  <Limit  WRITE>
    DenyAll
  </Limit>
</Anonymous>
```

El parámetro ServerName específica el nombre a ser desplegado cuando un cliente se conecta al servidor. Es desplegado antes del prompt de nombre de usuario contraseña. El ServerType específica si el servidor será ejecutado desde inetd o si será en modo de stand-alone. La directiva MaxInstances puede ser usada para limitar el número de conexiones que un servidor puede aceptar. Los parámetros de User y Group dan el nombre del usuario bajo el cual el servidor se ejcuta. Una vez el servidor se ha establecido y no necesita ya los permisos de root, este cambiará su identidad al usuario y al grupo aquí señalado. La sección Directory establece los parámetros que deben ser aplicados a un directorio en particular. En nuestro ejemplo, queremos que los usuarios puedan sobreescribir archivos y permitiremos eso en cualquier subdirectorio del directorio root.

La sección que contiene la palabra clave Anonymous establece el acceso de los usuarios anonimos, ejecutando como el usuario de nombre ftp. Hasta 10 usuarios anonimos son permitidos a la vez, usando o el nombre “anonymous” o el nombre “ftp”. La directiva Limit niega el permiso a los usuarios anonymous de subir o escribir archivos al servidor.

Ejercicio 7-3: Configurar los Servicios de FTP Usando ProFTPD

Las soluciones a este ejercicio se incluyen en su contenido.

1. Instale el paquete de ProFTPD. Si su distribución viene ya con el incluido en una versión pre-empaquetada use esa. Sino descargue el paquete desde el sitio web de ProFTPD http://www.proftpd.net, descomprimale, compilelo e instalelo.

2. Debe crear un directorio para el acceso anónimo:
```
# mkdir  /home/ftp
```

3. Edite el archivo /etc/proftpd.conf. Primero configure el servidor para solo tener acceso de usuarios autorizados:
```
ServerName              “FTP de CodigoLibre”
ServerType              inetd
DefaultServer           on
Port                    21
MaxInstances            30
User                    nobody
Group                   nobody
Umask                   022
<Directory  /*>
    AllowOverwrite      no
</Directory>
```

4. Edite el archivo /etc/inetd.conf. Elimine el simbolo de comentario de la línea que pertiene a cucalquier servicio de FTP actualmente. Agregue la siguiente línea:
```
    ftp   stream   tcp   nowait   root   /usr/sbin/tcpd   in.proftd
```
Usamos wrappers TCP aquí, wlo cual no es estrictamente necesario ya que ProFTPD puede efectuar control de acceso a los hosts por si mismo, pero ers bueno ser consitente y usar wrappers TCP en todos los servicios.

5. Le enviamos una señal al inetd para que lea nuevamente su archivo de configuración:
```
# klllall  -HUP  inetd
```

6. Pruebe el servidor ejecutando un cliente FTP. Es mejor provar desde un equipo diferente, siempre y cuando sea posible. Será necesario suplir un nombre de usuario y contraseña válidos en el sistema.

7. Habilite el acceso anónimo agregando las siguientes líneas al final de su archivo /etc/proftpd.conf:
```
<Anonymous   /home/ftp>
User                    ftp
Group                   ftp
UserAlias               anonymous ftp
MaxClients              10
  <Limit  WRITE>
    DenyAll
  </Limit>
</Anonymous>
```

8. Pruebe el servidor ejecutando un cliente FTP, desde un computador diferente si es posible. Ingrese (login) con el nombre de usuario "ftp". Se le pedirá en el prompt entrar su dirección de correo en vez de una contraseña.

SQUID

Si cada petición de un objeto Web requiriera una conexión al servidor actual efectuando el hosting y la recarga completa del objeto desde la data de origen, su navegación en el Internet fuese lentísima y frustratoria. Lo que hace falta es un método de capturar (caching) los objetos Web más frecuentes y recientes. Bueno este tipo de software ya existe y es conocido como caching proxy server (Servidores Proxy de Captura).

Un servidor proxy es un servidor transparente que sirve como mediador entre una red privada y el Internet. Los usuarios navegan vía el servidor proxy para accesar la información desde el Internet. Si un usuario hace una petición de una página Web en el Internet desde una red protegida, la petición irá al servidor proxy antes de ser pasada al servidor real. El servidor proxy entonces revisa para ver si la página requerida está en su cache. Si la página está en el cache del proxy, el servidor proxy le envia esta página directamente. hay funciones adicionales que efectuá un servidor proxy que serán explicadas en las siguientes secciones.

Hay diferentes métodos de configurar un servidor proxy. Una de las configuraciones más popular de configurar un servidor proxy es en la que tanto como las conexiones entre el Internet y la red protegida son atraves del servidor proxy. En esta instancia, un servidor proxy es nada más que un programa que reside en el firewall (corta fuego) y puede ver ambos lado de la interfaz: la intranet y el Internet. Aunque muy a menudo el firewall y el servidor proxy por lo general aparecen juntos, no son la misma cosa. El firewall filtrador de paquetes opera en la capa de Red del modelo OSI, mientras que los servidores proxy trabajan en la capa de Aplicación.

Un firewall filtrador de paquetes puede bloquear varios protocolos y direcciones IP de entrar a nuestra red local. Este también puede controlar los protocolos. Este también puede controlar los protocolos que los usuarios son permitidos tener acceso al conectarse al Internet.

Los proxies pueden ser clasificados especificos a su aplicación. Algunos ejemplos de servidores proxy son un proxy HTTP para las páginas Web, un servidor proxy FTP,un proxy SMTP/POP para los e-mails, un proxy NNTP (Network News Transfer Protocol) para los servidores de noticias, entre otros.

GNU/Linux incluye un proxy cache de páginas Web muy popular llamado Squid. En esta sección, cubriremos los siguientes tópicos:
- Configurar Squid
- Iniciar y Detener Squid en un Servidor GNU/Linux
- Características de Seguridad de los Servidores Proxy
- Función de Caching
- ACL (Access Control List/Listas de Control de Acceso)

Configurar Squid

Squid Puede ser configurado dependiendo de las necesidades de un site. El archivo de configuración de squid se encuentra en el directorio /etc/squid/. Algunas de las opciones comúnmente modificadas por el administrador en el arcivo de configuración se muestra aquí en lo adelante:

cache_dir Type Dirname Mbytes Level1 Level2

La función de esta opción es la de especificar el nombre del directorio en el cual se almacenará el cache.

pid_filename filename

Esta opción crea el nombre de archivo para almacenar el ID del proceso de Squid. Este archivo es usado por el script /etc/rc.d/init.d/squid para matar (kill) el proceso cuando se llama el parámetro de detener (stop).

ftp_user user@domain.name

Esta opción por lo general específica la dirección de correo electrónico del administrador del servidor. Esta opción es necesaria porque algunos sitios FTP requieren que el usuario pase una ID de usuario válida.

cache_mgr user@domain.name

Esta opción específica la dirección de correo electrónico del administrador del servidor. Se envia un e-mail al administrador del sitio si ocurre un problema con el cache.

cache_effective_user username

cache_effective_group groupname

Si squid es ejecutado como root, le cambiará en efecto el userlD o el groupID al nombre de usuario o al nombre del grupo especificado en esta opción.

Iniciar y Detener un Squid Ejecutando en un Servidor GNU/Linux

Como todo otro servicio de GNU/Linux, iniciar y detener el Squid es una tarea simple, con los siguientes comandos podemos dar inicio y detener el servidor desde la línea de comandos:

/etc/rc.d/init.d/squid start
/etc/re.d/init.d/squid stop

Después de iniciado, el Squid escucha en el puerto 3128 por defecto. Los paquetes necesarios para ejecutar squid son explicados brevemente en lo que sigue:

/etc/rc.d/init.d/squid	El script de inicio/detener (start/stop)
/etc/squid	El directorio de configuración
/usr/doc/squid-version/	Directorio de la documentación
/usr/lib/squid/	Directorio de los archivos de soporte
/usr/sbin/client	Programa de diagnóstico de línea de comandos
/usr/sbin/squid	Programa del daemon principal
/var/log/squid/	Directorio de log/diario
/var/spool/squid/	Directoriodel cache

Los ISPs por lo general proveen acceso atraves de un servidor proxy de Web. El navegador del cliente es normalmente configurado para que use el servidor proxy remoto. En la siguiente tabla se muestra la conexión del cliente local al potocolo del servidor remoto para el servicio de proxy de Web.

Descripción	TCP Protocol	Dirección Remota	Puerto Remoto	Entrada/ Salida	Dirección Local	Local	TCP Flag
Petición del Cliente Local	TCP	Servidor Web Proxy	Web Proxy Puerto	Salida	IPADDR	1024:6555	Any
Servidor Remoto	TCP	Servidor Web Proxy	Web Proxy Puerto	Entrada	IPADDR	1024:6535	Ack

El ipchains fué por mucho tiempo el programa más común de la administración de firewalls, hoy día ha sido reemplazado por iptables, pero para ilustrar ipchains funcioa igual. Usado para especificar reglas de filtrado de paquetes.

Primero, las variables de ambiente para del puerto proxyy el servidor proxy son asignadas:

ISP Servidor Web proxy.
WEB_PROXY_SERVER = “mi.www.proxy”
ISP puerto Web proxy. Por lo general se usa el 8080 o el 8008.
WEB_PROXY_PORT = “proxy port”

Ahora use el comando ipchains para definir las reglas del proxy Web. La primera regla de filtrado de paquetes es para la cadena de salida/output:

```
ipchains  -A  output  -i   $EXTERNAL_INTERFACE   -p tcp \
-s   $IPADDR  $UNPRIVPORTS   \
-d $WEB_PROXY_SERVER  $WEB_PROXY_PORT   \
-j  ACCEPT
```

Las opciones especificads son las siguientes:

-A output
Esta agrega la regla al final de la cadena de salida.

-i $EXTERNAL_INTERFACE
Esta específica la interface interconectadas, e.j., eth0 o eth1.

-p tcp
El protocolo IP al cual las reglas aplican es el TCP.

-s $IPADDR $UNPRIVPORTS
Esta específica la dirección origen del paquete y la dirección del puerto no privilegiado. En este caso, la dirección destino es su dirección IP.

-d $WEB_PROXY_SERVER $WEB_PROXY_PORT
Esta específica la dirección de destino del paquete como un valor en $WEB_PROXY_SERVER. El puerto Web proxy es el valor en $WEB_PROXY_PORT.

-j ACCEPT
Esta política le dice que acepte los paquetes.

La regla para la cadena de entrada se presenta aquí debajo:

```
ipchains  -A  input  -i  $EXTERNAL_INTERFACE  -p tcp  !
-y  -s  $WEB_PROXY_SERVER  $WEB_PROXY_PORT
-d  $IPADDR $UNPRIVPORTS  -j  ACCEPT
```

Es muy similar a la cadena de salida. la única opción agregada es la parte de ! -y. Esta opción establece la opción ACK en un mensaje TCP. La otra diferencia es que la dirección fuente o de origen y la de destino son exactamente opuestas.

Características de Seguridad de los Servidores Proxy

Un servidor proxy provee muchas características de seguridad, algunas de estas las discutimos en esta sección. Las peticiones para las URL remotas son capturadas por el servidor (HTTP) proxy y, si son permitidos por las reglas base del firewall, son pasadas al servidor de Internet remoto.

La data desde las redes externas hacia las redes protegidas, como es e-mail, es capturada por el servidor (SMTP) proxy y, si es permitido por la base de reglas del firewall, es pasada dentro de la red privada. Correos Basura/Junk o malicioso pueden ser bloqueados con este servidor.

Otra característica importante es “audit trial”. Esta características puede ser considerada como una de seguridad en una compañía cuando buscan por algún tipo de violación de políticas. De toda otra forma, auditar es por lo general hecho para encontrar sitios y para llevar acabo encuestas.

Las peticiones efectuadas al proxy son almacenads en el siguiente archivo del log:
/usr/local/squid/logs/access.log

La información como el número de personas que han usado el cache, cual página cada uno peticionó y cuales páginas son las más populares, es almacenada aquí.

La Función de Cache

Una de las funciones más importante de un servidor proxy es el cache de la data (cahing). Se establece un cache para almacenar una copia de los documentos navegados de Internet localmente, revisa si cualquier de los documentos almacenado ha cambiado y actualiza esos que si ha cambiado. La técnica de caching ahorra ancho de banda en pa red y el tiempo que se toma en buscar los objetos es reducido significativamente.

Aunque caching aumenta la velocidad de búsqueda debemos tener cuidado al especificar el contenido del cache. Por ejemplo, páginas que contienen la información de tarjetas de crédito nunca deben ser almacenads en cache.

En el ejemplo debajo muestra como un servidor Squid implementa caching. La palabra clave cache_host específica los servidores cache padre e hijos. Aquí le presentamos una porción del archivo squid.conf:

```
#  Squid.conf   - en el host  : childcache.codigolobre.org
#  El formato es: hostname    tipo   http_port     udp_port
#
cache_host   parentcache.codigolibre.org        parent 3128    3130
cache_host   childcache1.codigolibre.org        sibling 3128    3130
cache_host   childcache2.codigolibre.org        sibling 3128    3130
```

Al efectuar la búsqueda de las peticiones, el Squid primero revisa su propio cache, entonces busca a los siblings por la petición. Si ninguno de los host cache o ni los siblings (hermanos) tienen el objeto requerido, este le pide a uno de sus chache padres que lo busque desde source.

Lista de Control de Acceso (ACL)

El ACL es una especie de portero (gatekeeper) que permiten o previenen a otros de accesar a un host que también es usado por la jerarquía del cache como un servidor proxy. Para empezar, un ACL , es definida , entonces el acceso a una función del cache es permitido o denegado. En la mayoría de los casos, esta característica está en http_access y permite o niega a un navegador Web accesar a un host.

El Squid inica desde el principio de la lista y trabaja decendiendo al decidir en cual clase uno cae y si se le permite o se le dieniega el acceso. En el siguiente ejemplo, un usuario desea un red de Clase C (a/24) que consiste del rango de direcciones 192.1.2.0 to 192.1.2.255 para que tengan acceso a la Web atraves del proxy:

```
acl  hostspermitidos    src   192.1.2.0/255.255.255.0
acl  all   src  0.0.0.0/0.0.0.0
http_access  allow  hostspermitidos
http_access   deny  all
```

El /255.255.255.0 es equivalente al /24 en la regla de ipchains anterior (los primeros 24 bits de la dirección). El src en la primera línea es una de las opciones disponibles para determinar en cual ACL la persona se encuentra. Un usuario puede elegir cuales computadoras accesan el proxy por la dirección IP, fecha actual, el site destino, etc.

Supongamosno que una conexión desde 192.1.2.* está usando TCP y nos peticiona un URL. El Squid revisará el http_access línea por línea y se detendrá en la primera que iguala para determinar en cual ACL el host iiciando la conexión pertenece. El Squid luego procederá a permitir la petición. Es importante denotar que invertir las dos últimas líneas de comandos en el ejemplo anterior no funcionará.

```
acl  hostspermitidos    src   192.1.2.0/255.255.255.0
acl  all   src  0.0.0.0/0.0.0.0
```

http_access deny all
http_access allow hostspermitidos

Esto porque Squid no igualará en la primera línea http_access y denegará la conexión a todos los otros usuarios.

El squid.conf que viene por defecto contiene los siguientes parámetros por defecto:
acl Administración proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl all src 0.0.0.0/0.0.0.0
http_access deny Administración!localhost
http_access allow all

En la primera línea, el campo proto se refiere al bloqueo del indicado protocolo cache_object, pero HTTP o cualquier otro protocolo puede ser indicado. El protocolo cache_object es único al Squid y sólo retorna información al remitente acerca de la configuración del cache o el estado. El acceso es denegado a una conexión intentando usar el protocolo cache-object protocol (como está definido en el acl administrador) almenos que este es del acl localhost. De esta manera, ningún equipo de afuera puede accesar información acerca de el estado interno del Squid, mientras que un programa ejecutándose en servidor actual del cache si puede. Administración, pero no localhost, es denegado ya que "!" significa "no" . Las máquina cliente en cualquier red son permitidos acceso.

ACLs Basadas en Direcciones de Destino

El siguiente ejemplo de los parámetros de configuración muestran una aplicación popular y necesaria del Squid:
acl adultos dstdomain no-se-debe-ver.com
tampoco-se-puede-ver.com
acl hostspermitidos src 192.1.2.0/255.255.255.255.0
acl all src 0.0.0.0/0.0.0.0
http_access deny adultos
http_access allow hostspermitidos
http_access deny all

A menudo deberemos tener que denegar acceso a una lista de sites inapropiados. Aunque Squid no está optimizado para efectuar esto para un gran número de sites, por lo general puede manejar un número de sites que si es suficiente. El este ejemplo no referimos al bloqueo de los hosts peticionando URLs que van a no-se-debe-ver.com o a tampoco-se-puede-ver.com ya que estas igualarán la primera línea de http_access cuando el Squid revisa la listas de http_access. Todos los hosts que hacen peticiones de sites de adultos serán denegados. Si el host pasa esta regla, ellos entonces son revisados para deteminar si ellos son del rango de dirección IP correcto, si es así, ellos son permitid acceso. Si el host no esta en este rango de direcciones, entonces iguala el acl all y iguala la última regla http_access. Asi pues, ellos son denegados el acceso.

Squid es por lo general iniciado desde un scrip del rc al inicio del sistema; asi que es ejecutado como root. Pero, no es necesario ejecutar a squid como root y usted debe crear un grupo y llamarlo squid y que conternga un sólo usuario de nombre squid, el squid auromáticamente se cambiaría es este usuario no privilegiado después de adherirse a puerto de la red.

Ejercicio 7-4: Instalar y Compilar Squid

Para esta práctica necesita tener instalado un versión del compilador gcc y un ambiente de build válido. Además deberá estar ejecutan X, con un Mozilla ya instalado, para poder provar el squid al menos que

sepa navegar en lynx. Preparecese para describir que va sucediendo. ¿Dónde se instaló el software? ¿Dónde están los binarios del squid? ¿Dónde esta el archivo squid.conf? Elimine cualquier copia adicional de squid.conf y debe crear un viínculo simbólico a /etc/squid/squid.conf en su lugar. Note que muchas distribuciones incluyen a Squid en sus instalaciones. Usted puede revisar si la suya es una usando el comando ls -R /etc | grep squid para ver si Squid ya está instalado. Un resultado similar al siguiente indicaría que si ya esta instalado:

```
$ ls -R /etc | grep squid
squid/
squid
squid*
/etc/squid:
squid.conf
```

Fijese que los resultados incluye los directorios pero no necesariamente las localidades. La línea squid* es el script de ejecusión, que se encuentra en /etc/rc.d/init.d/ para esa distribución en particular.

Puede ser que elija no efectuar los pasos del 1-6 si usted ya tiene instalada una versión de Squid, aunque algunas cosas en específico puede que cambie dependiendo de su distro. Esta información es lo más probable bueno para la mayoría de situaciones pero puede ser muy importante si usted no tiene un ambiente vealido de gcc. Las soluciones a este ejercicio se encuentran en el Apéndice A.

1. Dirijase al Internet y descargue la última versión estable del Squid.
2. Descargue el fuente y descomprimalo en el directorio apropiado.
3. Siempre lea los archivos de instrucción INSTALL, Readme y QUICKSTART.
4. Proceda a configurarlo.
5. Ejecute el make all para compilar el software.
6. Ejecute la sentencia make install para instalar el software. Deberá ser root para instalar el softeare por cuestión de permisos.
7. Edite el archivo squid.conf para modificar los valores apropiados para las siguientes variables. Puede ser que solo tenga que modificar uno o dos de estas variables.

   Cambie el parámetro http_access de denegar a aceptar (deny a allow). Usted podrá luego refinar estos parámetros para restringir acceso a redes, hosts y usuarios en específico.

   ```
   http_access allow all
   ```

   Puede que tenga que modificar para especificar un servidor DNS si la función ya no se encuentra ejecutándose en su sistema. Deberá quitar el comentario a la línea “dns_nameservers” y reemplazar “none” con la dirección IP apropiada:

   ```
   dns nameservers 192.168.2.1
   ```

   Si desea hacer que la notificación por e-mail funcione deberá especificar el correo electrónico del administrador. Elimine el comentario de la línea “cache_mgr” y reemplace Webmaster con la dirección de correo apropiada:

   ```
   cache_mgr admin@codigolibre.org
   ```

8. De inicio al squid con el parámetro -z para crear la estructura de directorio de cache. Esto puede llevar a cabo por varios minutos una actividad masiva en sus discos.

9. De inicio al squid una segunda vez, ahora sin la opción -z, y veríﬁque que el proceso eestá ejecutándose. Recuerde que squid requiere acceso al DNS (o localmente o un host en otra red) para operar.

10. Configure su navegador para que apunte al Squid.

11. Verifique que su navegador estasu navegador está funcionando a traves del Squid accesando una página Web. Usted puede verificar que lapágina está pasando por el Squid, usando el comando tail para leer el archivo /var/logs/access.log. Si no le registra, entonces verifique su archivo squid.conf, en particular los parámetros en el http_access.

NOTICIAS

En esta sección, cubriremos el uso del Sistema de Noticias Usenet. Una de las primeras aplicaciones del Internet (en aquel tiempo ARPANET) fué la transportación de noticias. La habilidad de poder tener grupos de noticias, listas de noticias y otras colecciones de informaciones permitiá a los usuarios del Internet mantenerce en contacto uno con otros. GNU/Linux implementa servicios y programas para darle a los usuarios una conexión a los grupos de noticias y servicios disponibles en el Internet. En particular, cubriremos el protocolo NNTP, el INN y aplicaciones CNews.

Los siguientes tópicos son discutidos en esta sección:
- Usenet
- NNTP (Network News Transfer Protocol)
- INN (InterNetNews)
- CNews
- Learnode
- Los Lectores de Noticias (Newsreaders)

Usenet

El Usenet es un sistema distribuido de noticia, usado para transmitir artículos con varios tópicos de discusión. Los tópicos son organizados en grupos de noticias (newsgroup) y el grupo de noticias es organizado en una jerarquía. Cuando te suscribes a un grupo de noticias particular, recibes un listado de todos los artículos de el servidor de noticias. Luego puede seleccionar un artículo para descargar desde el servidor.

El grupo de noticias Usenet trabaja muy parecido a una lista de correo, pero por su naturaleza de distribución es más eficiente para grupos grandes. Además, el mecanismo de distribución y protocolos usados son diferentes. Los grupos de noticias no han recibido una gran atención recientemente, debido a la concentración del HTTP. Esto es malo porque las noticias Usenet son mejores y más eficientes para trabajar las discusiones distribuidas.

Los grupos de noticias Usenet son organizados en una jerarquía, con diez categorías principales definiendo todos los tópicos como se lista en la siguiente tabla:

Categoría	Tipos de Tópicos
alt	Cualquier cosa
biz	Asuntos relacionados con Negocios
comp	Hardware de Computadoras, software y soporte a usuarios
humanities	Las Artes, Literatura y Filosofía
misc	Misceláneos, Empleos y Salud
news	Información de las noticias Usenet
rec	Recreación, Deporte y Pasatiempos
sci	Ciencias, ambas aplicadas y social

soc	Asuntos Sociales y Culturales
talk	Discusión y debates pertinente a la actualidad

La jerarquía Usenet puede ser comparada a la del DNS pero leyendo de izquerda a derecha. Un grupo de noticias sobre el tópico de seguridad de Linux puede ser comp.os.linux.security, aunque los tópicos similares pueden ser posteados en un grupo inferior a la jerarquía comp.securty. Las regiones geográficas también definen grupos de noticias para discusiones locales o regionales.

NNTP (Network News Transfer Protocol)

NNTP es un protocolo usado para suscribirse a varios grupos de noticias. NNTP no es un paquete de software particular, pero es un estándar de Internet. El está basado en una conección mediante un cliente en cualquier punto de una red y en un servidor que mantiene sus noticias en un disco de almacenamiento. NNTP provee un número de comandos convenientes que permiten al usuario descargar la cabecera o el cuerpo de un artículo por separado o líneas únicas de cabeceras desde una variedad de artículos. Esto permite que todas las noticias sean guardadas en un host central, con usuarios en la red usando programas clientes basados en NNTP para leer y postear. Hay muchos paquetes NNTP disponibles. El más usado es el Demonio NNTP o nntpd. El paquete NNTP contiene un servidor y dos clientes para descargar y publicar noticias.

Instalación

El servidor NNTP, llamado nntpd, puede ser compilado en dos formas dependiendo en la carga esperada en el sistema. No hay versiones compiladas disponibles porque algunos por defectos del sistema son codificados dentro del ejecutable, nntpd puede ser ajustado para que se inicie desde el script rc.inet2 o como un demonio administrado por inetd. Para el demonio, debes tener la siguiente entrada en /etc/inetd.conf:

nntp stream tcp nowait news /usr/etc/in.nntpd nntpd

Si se desea configurar como un stand-alone, tiene que estar seguro que esta línea esté comentada en el archivo /etc/inetd.conf. Pero para ambos caso, la siguiente línea tiene que estar presente en el archivo /etc/services:

nntp 119/tcp readnews untp #Network News Transfer Protocol -NNTP

Para almacenar temporalmente cualquier artículo entrante, debe existir un directorio .tmp en su cola (spool) de noticias. Este debe ser creado así:

```
# mkdir  /var/spool/news/.tmp
# chown  news.news  /var/spool/news/.tmp
```

Restringiendo Acceso NNTP

En el archivo nntp_access, usted puede capitalizar fichas (tokens) como xfer o read. Esto hará que nntpd requiera autorización desde el cliente antes de que cualquier acción pueda tomar lugar. Por ejemplo, para que un cliente se pueda conectar a un servidor NNTP, el dominio del cliente tiene que ser listado en el archivo nntp-access. Una vez que este acceso ha sido concedido, se pueden hacer restrinciones que permitan al cliente leer (solamente) o para leer correo transferido (para otro nodo del servidor de correo) o poder restringir la habilidad para postear a grupos de noticias.

Los derechos de acceso al NNTP son ajustados en el archivo /usr/lib/news/nntp_access. Las líneas del archivo muestran derechos de acceso otorgados para los hosts foráneos. Cada línea tiene el siguiente formato:

site read| xfer | both | no post | no [!exceptgroups]

Cuando un cliente se conecta al puerto NNTP, el servidor intenta obtener el nombre del dominio del host.

Esto entonces es igualado con el campo site en la línea anterior. La igualdad puede ser establecida completa o parcial. Si es completa, se aplica. Cuando sólo es parcial, se aplica si ninguna otra conexión iguala. Una vez se iguala, el segundo y tercer campo dicta los derechos de acceso otorgados al cliente. El cuarto campo es opcional y contiene una lista de gupos, separados por coma, que el cliente es negado el acceso.

Temas de Seguridad NNTP

Los Crackers regularmante escanean el Internet buscando máquinas que soporten el servicio NNTP. Ellos buscan máquinas que ellos puedan leer o postear mensajes anónimamente. Hay sitios Web que listan servidores NNTP abiertos que puedan ser usados anónimamente. El demonio INN procesa mensajes de control de newgroup y rmgroup en un shell script que usa el comadno del shell eval. Algunas informaciones pasadas a eval vienen del mensaje sin la revisión adecuada de caracteres que son especiales para el shell. Esto permite que cualquiera que pueda enviar mensajes a un servidor INN pueda ejecutar comandos arbitrariamente en el servidor y lo que es obviamente riesgos de seguridad.

INN (InterNetNews)

Incluida en la mayoría de versiones de GNU/Linux esta InterNetNews (INN). INN es una implementación del NNTP (Network News Transfer Protocol) y es capaz de manipular grandes cargas de noticias y más sistemas complejos de noticias. Aunque GNU/Linux es proveído con una versión de INN, debido al factor de que INN es usado para grandes cargas de noticias y con sistemas de noticias complejos, no es muy usado en GNU/Linux. Si INN va ha ser usado en un sistema GNU/Linux para un sistema grande y complejo de noticias, la configuración de INN se torna un poco complicada. Para una LAN o un sistema stand-alone que usa GNU/Linux, INN es más fácil de configurar.

INN incluye dos demonios: innd y nnrpd. El innd sirve para el mismo propósito como nnrpd, pero permite múltiples cadenas de transferencia a la vez. El nnrpd maneja los servicios de newsreader.

CNews

El otro agente de transporte popular es conocido como CNews. Apropiadamente, CNews es el viejo roble en el mundo de noticias de UNiX, desde los tiempos de interconexiones modems-a-modems a 300 bps y uucp. El desarrollo de una variedad de programas de noticias se inició con ANews y ha evolucionado hasta CNews.

CNews requiere partes del NNTP para poder funcionar correctamente, asi reduciendo más su necesidad. Aún asi, puede ser que un día se enfrene con un servidor UNiX desactualizado que aún este ejecutandolo, asi que es bueno conocer acerca de CNews.

Instalar CNews es un ejercicio simple de seguir las siguientes instrucciones. Ya sus instrucciones han sido bien depurads. Configurar CNews es la simple interacción de tres comandos: cron, addgroup y addfeed.

La configuración de CNews se basa en el colocamiento apropiado de los diferentes componentes. Los archivos de configuración se encuentran en /usr/lib/news. La mayoria de los binarios están en el directorio /usr/lib/news/bin. Adicionalmente, los artículos de noticias están localizados en /var/spool/news. Como los diferentes archivos están separados, un importante recordar que los diferentes directorios son propiedad del usuario news y el grupo news y son configurados individualmente. Pero, la excepción es setnewsid, la cual es la propiedad de root y maneja los IDs reales de usuarios en ciertos programas de correo. La mayoría de los problemas con los usuarios nace en la incapacidad de CNews de accesar los archivos de los usuarios.

Entregando las Noticias

Hay dos técnicas para la entrega de noticias atravez de CNews, dependiendo en donde el artículo de noticia se originó. Cuando un usuario local postea un artículo, el comando inews maneja el artículo, después el newsreader (lector de noticias) lo enruta al comando. El uso de inews es necesario para comple-

tar la información de cabecera. Si la noticia proviene desde una localidad remota, los artículos son pasados al comando rnews para almacenarlos dentro del directorio /var/spool/newsin.coming. Luego, los artículos serán recogidos por newsrun. Como quiera que llegen los artículos de noticias, eventualmente serán manejados por el comando relaynews para el enrutamiento y otras funciones administrativas.

El comando relaynews ejecuta el volumen de la administración del enrutamiento de los artículos en CNews. Cuando un nuevo artículo de noticia llega, relaynews revisa a ver si el artículo ya existe en el site local buscando en el historial o el ID. Una vez que esto se haya cumplido correctamente, relaynews revisa los Newsgroups: la línea cabecera para ver si el grupo que originó el artículo es requerido por cualquiera de los sites locales. Si un site local hiso una petición del artículo y el Newsgroup está listado dentro del archivo activo, relaynews almacenará el artículo en el area de noticias correcta. Si el directorio no existe, lo crearia para almacenarlo. En cualquiera de los casos, el ID del mensaje será escrito al archivo de historial.

Si relaynews revisa el Newsgroup: la línea cabecera y encuentra que el Newsgroup no esta localizado dentro del archivo activo, el artículo sería movido a un grupo basura para almacenarlo temporalmente.

Al tratar de enviar noticias a un site UUCP remoto, ya sea un artículo o una batch (por lote), uux es usado para activar y ejecutar el comando rnews localizado en el site remoto. Los artículos son entonces alimentados al usuario remoto en la entrada estandar.

Si un sitio activa los archivos por lote o batch, CNews usa el mismo método para enviar múltiples artículos de noticias como lo hace para mandar artículos únicos de noticias. En lugar de enviar el correo inmediatamente, CNews cambia la ruta para el archivo out.goig/site/togo.

Instalación

La distribución GNU/Linux Debian es la única distribución que viene con CNews en el CD, pero los RPMs están disponibles.

Una vez los archivos esten colocados dentro de los directorios correctos, el archivo de configuración que se encuentra en el directorio /usr/lib/news, o en algunos sistemas en /etc/news, tiene que ser editado. Un sistema de Newsgroup puede ser configurado con la copia de un sistema existente del site que el recibe noticia. Establesca la propiedad de este sistema a un grupo y establesca el modo a 644 una vez el sistema ha sido colocado apropiadamente en /usr/lib/news o /etc/news. Elimine cualquier grupo que se inicia con “to.” Esto incluye los grupos to.localsite y to.remotesite, donde localsite es su site local y remotesite es el site donde los correos serán enviados.

Configuración de Cnews

Como mínimo, necesitará editar o cuando menos echar un vistazo a los siguientes ficheros. Todos deberían estar en /usr/local/lib/news:

- **active** - el fichero con los grupos activos
- **batchparms** - parámetros de proceso por lotes
- **explist** - configuración de caducidad de los artículos
- **mailname** - nombre de su servidor
- **mailpaths** - direcciones de los moderadores de grupos
- **organization** - su organización
- **sys** - control de lo que se pide y envía
- **whoami** - el nombre de su sistema para la línea Path.

El próximo paso es editando este comando:

```
$ cp active active.old
$ sed 's/ [0-9* [O-9]* / 0000000000 00001 /' active.old > active
$ rm active.old
$ cd /van/spool
$ mkdir news news/in.coming news/out.going
$ chown R news.news news
$ chmod -R 755 news
```

Esto completa la instalación de CNews.

Leofnode

Como puede leerse en la página del manual (man fetch) leafnode es un paquete de trabajo con news diseñado para ordenadores pequeños con pocos usuarios, de manera que no ocupa mucho espacio y permite la gestión de muchos grupos. El diseño de leafnode está pensado para que sea capaz de autoreparar errores cuando ocurran, y que no necesite mantenimiento manual. Es decir: nos encontramos ante un programa que requiere poco espacio y mantenimiento, pero que en cambio nos proporciona total acceso a la gestión de grupos de noticias, gracias a leafnode (un servidor de NNTP), fetch, el programa que se encarga de recoger las news de Internet y dejarlas en nuestro disco duro para su posterior lectura por parte del programa que deseemos, y texpire, que se encarga de eliminar los mensajes viejos para recuperar disco duro y deshacernos de los artículos no deseados según el tiempo de expiración que hayamos configurado.

Leafnode es un programa incluido en la mayoría de las distribuciones GNU/Linux (sobre todo en las de RedHat), y si no disponemos de él puede encontrarse en ftp.redhat.com en el directorio pub/manhattan/contrib/i386 (versión 1.5) o en la sección de viejos RPMs (la versión 1.4). Algunas (caso de algunas distribuciones Manhattan) versiones tienen un fallo con Netscape, pero al autor le consta que a partir de la versión 1.9 este fallo fué corregido completamente (NOTA: El autor usa leafnode 1.4 ya que le sigue funcionando perfectamente y no ve ninguna necesidad de actualizar a la version 1.9 :). La instalación en formato rpm es mediante el comando rpm -i, y para paquetes .deb y tar.gzs también se debe seguir el método habitual de instalación ya conocido por los que usan distribuciones Debian o Slackware, además de poder convertir el paquete de rpm a cualquier otro formato mediante el script alien, incluido en muchas distribuciones y que podemos encontrar en el home de Debian.

Lectores de Noticias

Los grupos de noticias una vez fueron los medios primarios de comunicación en el Internet. Con la llegada del WWW, esto ha cambiado. El uso de los grupos de noticias se han convertido en una área especializada. Una excepción es soporte computacional, para el cual los grupos de noticias aún proveen una exelente fuente de información.

Navegadores Web modernos acompañan sus clientes de correo con un lector de noticias perfectamente funcional. Muchos de los programas disponibles son clones uni del otro y no nesecitan ser mencionados aqui. Mencionamos lectores basados en texto que proveen un exelente funcionamiento y la habilidad para explorar los Newsgroups desde una consola de texto. Para una lista exaustiva, ver:

http://www.newsreaders.com/unix/clients.html.

No existe un "lector de noticias verdadero". Como resultado, hay muchos lectores bien conocidos que se compilan fácilmente bajo Linux en particular. En el momento de escribir esto, "tin", "trn", y "nn" están en la mayoría de las distribuciones de Linux disponibles y en newspack.

A la hora de elegir un lector de noticias, se requiere básicamente algo que sea fácil de usar, muy configu-

rable por el usuario, y con capacidad para ordenar por hebras y filtrar los artículos (seleccionar los artículos interesantes o hacer que los no interesantes no aparezcan en absoluto).

Puede configurar sus rutas como quiera en tanto en cuanto todos los lectores puedan encontrar "inews" de su instalación de Cnews o INN, y el programa "mail" para mandar respuestas por correo electrónico a los artículos.

trn/Mthreads

El trn es un derivado de "rn" con capacidad para ordenar en hebras. Los trn3.2 y superiores tienen la posibilidad de seleccionar entre "mthreads" (el creador de hebras de trn) o NOV (creador de hebras de INN). Escrito en Perl.

Para compilarlo, simplemente ejecute Configure y acepte las opciones por defecto. Es posible que necesite lib4.4.1 y bash-1.13 (hay una beta disponible en varios servidores de archivos de Linux) para ejecutar Configure con éxito. Probablemente necesitará tanto bash1.13 como libs4.4.1 para conseguir que el nuevo Configure funcione correctamente.

La distribución newspak de sunsite contiene ficheros de configuración funcionales para trn bajo Linux.

Es probablemente poco deseable intentar editar un config.sh a mano, a menos que esté Vd. haciendo algo *MUY* sencillo, como cambiar las rutas para adecuarlas a sus gustos. Si hace esto, necesitará ejecutar "Configure -S" antes de "make depend", "make" y "make install".

Aunque "Configure" falla generalmente bajo Linux con bash1.12, "Configure -S" funciona bien, así que si toma Vd. el config.sh de newspak como punto de partida, estará muy cerca.

Compilar con soporte NNTP es tan simple como contestar "yes" cuando Configure le pregunte si lo quiere (suponiendo que Configure funciona en su sistema). Un futuro lanzamiento de newspak incluirá un config.sh para NNTP, así como el ya existente para sistema local para aquellos de nosotros que seguimos usando bash1.12.

He usado trn sobre un enlace SLIP como lector NNTP. Si conecta Vd. con un sistema que disponga de todos los grupos de Usenet, se hará muy viejo esperando a que trn se baje el fichero "active" y a que ordene los artículos.

Hay docenas de parámetros con los que invocar a trn para obtener toda clase de comportamientos. Lea la página de manual de "trn" para conocer los detalles. Yo uso una buena prestación de trn para indicar todos los parámetros fácilmente:

- Cree un fichero ~/.trnrc con todas las opciones
- export TRNINIT="~/.trnrc"

La distribución actual de newspak tiene una copia de mi .trnrc como ejemplo.

trn3.2 y superiores permiten elegir al usuario hebras NOV o de mthreads. Por lo tanto, recomiendo compilar el programa para permitir ambos mecanismos (es una pregunta de Configure). Para elegir uno de ellos en el momento de ejecución, pruebe "trn -Zo" para NOV y "trn -Zt" para mthreads. Yo defino un alias para trn para usar el mecanismo deseado.

Para construir la base de datos de mthreads, haga algo como lo siguiente en el crontab de "news":

crear la base de datos de mthreads

35 * * * * /usenet/bin/mthreads all

slrm

El slrn es un newsreader que es fácil de leer basado en ncurses (menus, etc) que puede desplegar color, lo que es muy útil para navegar entre muchos correos. Si usted prefiere una interface visual a compensación de poder, este es su newsreader de consola. También funciona muy bien en consolas de X.

tin

Tin es un lector con capacidad de ordenamiento en hebras que trata de ser fácil para los nuevos usuarios. Soporta filtrado de artículos y hebras NOV. Si utiliza Vd. INN, leerá los ficheros .overview por defecto, y no escribirá ficheros índice.

Para compilar Tin bajo GNU/Linux, simplemente edite el fichero Makefile para indicar dónde se encuentran ciertos programas (especialmente la ubicación de inews) y teclee “make linux”. No se requiere ningún parche para tin bajo GNU/Linux.

Para ordenar los artículos en hebras, puede simplemente invocar a tin con el parámetro “-u” para actualizar los ficheros índice.

Para usar la capacidad de tin para leer vía NNTP, compílese con “NNTP_ABLE” definido. Esto resultará en un fichero llamado “tin” para el servidor local y otro llamado “rtin” para lectura NNTP. “tin -r” obtendrá el mismo resultado que “rtin”.

Iain Lea recomienda la siguiente entrada de crontab, y dice que se necesita hacer un “make daemon” para compilar tind.

```
# ordenar en hebras
35 * * * * /usenet/bin/tind -u
```

He usado tin sobre un enlace SLIP como lector NNTP. Si conecta Vd. con un sistema que disponga de todos los grupos de Usenet, se hará muy viejo esperando a que tin se baje el fichero “active”.

SINCRONIZACION DE TIEMPO

En esta sección, cubriremos la sincronización del tiempo usando el NTP (Network Time Protocol). Puede ser que la sincronización del tiempo no le paresca del todo importante al principio. Pero, hay muchas instancias donde si es importante. Es esencial cuando tienes que sincronizar cualquier base de datos distribuida. Es además requerido por algunos sistemas de autentificación, tal como Kerberos. Reduce la confusión creada cuando envias a alguien un mensaje de correo electrónico que aparenta haber sido enviado desde el futuro.

Los siguientes tópicos son discutidos en esta sección:
- NTP
- xntpd

NTP

NTP es el protocolo usado para sincronizar el tiempo entre máquinas. Las matematicas y comunicaciones involucradas en este protocolo diriamos son complejas. Básicamente, la configuración de los sistemas involucrados determina cuales sistemas deben ser confiados y que exactitud debemos asumir que ellos poseen.

Además el protocolo intenta determinar la latencia de la red o que tiempo un paquete de red para transportarse de un sistema a otro. El usa esta información para interpolar el tiempo actual. La exactitud resultante depende de la velocidad y consistencia de el medio de comunicación pero esta usualmente cae alrededor de unos cuantos milisegundos. Toda comunicación y calculos se hace usando el UTC (Universal Coordinated Time). Cada sistema puede entonces aplicar la información de la zona de tiempo que se encuentra para deducir el tiempo local.

El protocolo NTP usa un modelo cliente/servidor, con la capacidad de que servidores puedan actuar como peers para negociar tiempo común entre ellos. Los servidores colocados de una manera jerárquica. En la capa superior se encuentran los sistemas conectados directamente a una fuente de tiempo fisica. Este puede ser un reloj atómico o un receptor GPS, ambos suelen ser costosos. El reloj atómico dificilmente lo pueda adquirir una compañia!. Asi que los servidores de la capa (top tier) superior por lo general solo proveen sus servicios a servidores de la segunda capa. Estos servidores pueden entonces proveer el tiempo a sistemas clientes o para aún otra capa de servidores.

La version 3 es la actual de NTP, la cual es compatible con las versiones anteriores. Está definido como un estandar de Internet en el RFC 1305. Hay una versión 4 de NTP disponible, pero la especificacion no esta completa en este momento. La especificación y referencia de implementación están disponibles en el stio Web http://www.ntp.org.

NTP usa el puerto UDP 123 para comunicarse entre clientes y servidores. En este caso, el cliente contacta el servidor. Además tiene modos broadcast y multicast. En estos casos el cliente escucha los servidores para que envien información de sincronización del tiempo. La dirección IP multicast 224.0.1.1 ha sido asignada para ser usada por NTP.

xntpd

El paquete estándar usado en GNU/Linux para implementar NTP es xntpd. Es una implementación completa de la especificación de la version 3 del NTP. Además retiene ccompleta compatibilidad con servidores ejecutando las versiones 1 y 2. El paquete contiene un demonio que es usado para mantener la sincronización del tiempo y varias otras utilidades. El paquete xntpd fué derivado de una implementación de referencia llamada ntpd y el nombre fué cambiado para mostrar que el fué un versión de experimento. Increíblemente el nombre permaneció después que pasó del nivel experimental. Su nombre original de ntpd retornará con el lanzamiento de la versión 4.

El paquete xntpd está disponible en el sitio Web de NTP, http://www.ntp.org. Además podrás encontrar mucha información y documentación sobre NTP. La versión de desarrollo de ntpd versión 4 también está disponible.

Configuración

Configurar los ajustes de xntpd es muy simple. Nesecitarás una fuente de tiempo disponible. Hay varias disponibles de las cuales elegir. Primero, puedes conectar el sistema a una fuente de tiempo, como un GPS o un receptor FM que toman el tiempo desde un broadcast de radio. Por lo general, se toma el tiempo del Internet desde un servidor público de tiempo. Hay una lista localizada en http://www.eecis.udel.edu/-mills/ntp/servers.html. Si tienes más de dos sistemas para sincronizar, debes ajustar un sistema para sincronizarce con el servidor público y los otros sistemas sincronizarlos con el sistema que ya se ha sincronizado.

La configuración de xntpd se hace en el archivo /etc/ntp.conf. Una vez que se obtenga el tiempo atravez del servidor, simplemente ponga una línea en el archivo de configuración con el nombre del servidor procedido por el servidor principal. Por ejemplo, si vas a usar el servidor de tiempo time.mit.edu, deberás usar la

siguiente línea: **server time.mit.edu**

Usted puede especificar múltiples líneas de servidores. Cada servidor puede ser consultado y el tiempo que es usado será computado usando la información desde todos los servidores. Las lecturas más precisas serán dada la mayor importancia, asi que mientras más servidores usas, mejor y más exacto será su tiempo.

Debes además crear un archivo drift. Esto permitirá al demonio salvar parámetros entre los renicios para asi saber que tan exacto su reloj del sistema es. Para hacer esto, debes añadir la siguiente línea:

```
drift.file   /etc/ntp.drift
```

Si estas usando un reloj de referencia externo o no tiene acceso al Internet, deberás usar una pseudo dirección para el servidor. Cada reloj externo tiene su propia pseudo dirección y parámetros. El reloj interno de tu sistema tiene también una pseudo dirección, 127.127.1.0. El uso del reloj interno de la computadora no es recomendado. No debes usarlo si tienes la máquina en el Internet porque no debe desea propagar un tiempo incorrecto e inexacto. Además de la línea del servidor, nesecitarás proveer alguna otra información acerca de un reloj de referencia. Esto se hace usando la directiva fudge. Para el reloj interno, defina un valor estratum (usualmente un poco, pero menor de 15) y un ID de referencia. Lo siguiente es un ejemplo desde el archivo /etc/ntp.conf en un servidor NTP en una red aislada (stand-alone):

```
server  127.127.1.0
fudge  127.127.1.0  stratum 10  refid  time
```

Solo debes usar un servidor de tiempo externo o un servidor de referencia de reloj interno en una máquina en la red. Todas las otras máquinas deben usar ese servidor para sincronizar su tiempo.

Utilidades

Si por alguna razón no puedes ejecutar un demonio xntpd en su red, podrá actualizar su reloj del sistema desde un servidor NTP usando un modo directo. Para hacer esto, use el programa ntpdate, colocando el nombre del servidor a usar como argumento:

```
# /usr/sbin/ntpdate    time.mit.edu
```

Esto método no provee las ventajas que la sincronización intermitente. De cualquier manera, es muy útil si no tienes una conexión permanente al Internet.

Para revisar su demonio NTP, use el utilitario xntpdc. En el prompt que esta utilidad provee, si escriba “peers” le mostrará la información acerca de todos los servidores que el demonio ha contactado. La columna etiquetada “st” muestra el estratum o a que distancia del sistema está la fuente actual de tiempo. Si el número estratum es 16, significa que su sistema no ha sincronizados con el servidor.

Ejercicio 7-5: Sincronización del Tiempo de la Red

En este ejercicio, usaremos NTP como un cliente y como un servidor. Nesecitaras un sistema conectado al Internet sin un firewall de por medio para realizar la mayor parte de este ejercicio. Si no puedes tener acceso a un sistema con acceso directo al Internet, puedes saltar al paso 3 y cambiar la línea “servidor” en el paso 4 a las siguientes dos líneas:

```
server  127.127.1.0
fudge   127.127.1.0  stratum 10  refid  time
```

Las soluciones a este ejercicio están incluidas en el contenido.

1. Instale el paquete NTP. Si su distribución viene con una versión pre-enpaquetada, usala. De lo contrario, deberás descargar el fuente de ntp.org, desempaquetarlo, compilarlo e instalarlo. Note que el paquete

puede ser llamado por varios nombres. En Red Hat, es llamado xntp3, pero puede ser llamado ntp, ntpd, xntp o xntpd.

2. Si ya tienes un archivo /etc/ntp.conf, hasle un copia como copia de resguardo y eliminalo.

3. Ejecuta ntpdate a un servidor de tiempo público. Podemos sugerir como servidores de tiempo a ntp.css.gov, time.apple.com o time.mit.edu. Una lista completa de servidores de tiempo esta disponible en http://www.eecis.udel.edu/~mills/ntp/servers.htm.
   # /usr/sbin/ntpdate ntp.css.gov

4. Configure ntpd ahora como un cliente usando el mismo servidor de tiempo público. Debes crear el archivo /etc/ntp.conf que contengan sólo las siguientes líneas:
   server ntp.css.gov
   driftfile /etc/ntp.drift

   Usted puede especificar múltiples líneas de servidores si le gustaría cuestionar múltiples servidores. Esto provee redundancia si un servidor esta fuera y aprovecha mejores niveles de exactitud.

5- Inicie el demonio ntpd. (Recuerde que algunas distribuciones llaman al ejecutable xntpd.)
   # /usr/sbin/ntpd

6. Espere unos cuantos minutos para permitir al demonio sincronizarse. Usa el programa xntpdc para verificar que todo esta corriendo apropiadamente. En el prompt xntpdc, usa el comando peers:
   xntpdc > peers

remote	local	st	poll	reach	delay	offset	disp
ntp.css.gov	192.168.0.1	2	64	1	0.04797	0.00712	15.8750

   Si “st” es 16, entonces no estas sincronizado con el servidor de tiempo. Espera unos minutos o intenta con otro servidor.

   Corra ntpdate en otro sistema para asegurar que puedes tomar el tiempo y ajustarlo exactamente desde el servidor NTP.

PROTOCOLO RPC

Un mensaje enviado usando el protocolo RPC (Remote Procedure Call) es llamado una llamada RPC. El RPC transporta el comando acompañado con la información pertinente a la respuesta. Esta información incluye el tipo de archivo o respuesta retornada y la medida esperada. Todos los parámetros e información de llamada transferidos usando RPC son codificadas en un lenguaje llamado XDR (External Data Representation); pero, la información actual retornada no es codificada. El primer valor en un RPC (ambos envio y respuesta) es el XID. El XID es un identificador único que relaciona a esa llamada en específico y su respuesta. El XID identifica una llamada específica, lo cual permite al servidor determinar facilmente si una respuesta fué exitosa. Si fué exitosa, no son necesarias respuestas addicionales, ni en una retransmisión de la misma llamada. En el caso de una retransmisión debido a retraso de la red, este método previene que la llamada sea respondida múltiples veces. El XID sólo es parte de la cabecera que acompaña un RPC.

Una cabecera RPC contiene ocho campos antes a los parámetros de la respuesta esperada. El primero es el XID, seguido por un campo que contiene solamente un valor entero que identifica su protocolo transporte. El valor para un RPC siempre es 0. El tercer campo contiene información de la versión del protocolo. Actualmente hay solo una versión de RPC disponible, la versión 2. Los campos seguidos del número de versión contienen el número del programa, el número de la versión del programa y el número del procedimiento que debe ser llamado en ese programa. Los campos siete y ocho contienen información de autentificación para la llamada. El primero es el credencial seguido por el verificador. El credencial es usado para validar el cliente al servidor y el verificador es usado para validar el credencial. Esto es importante para hacer que la

información correcta llegue al cliente correcto.

La cabecera RPC es importante ya que gobierna el estado de la transferencia. Solo si las llamadas pasan ambas el examen de la versión del ROC y la de autentificación es que la respuesta traerá la información pedida; de cualquier otra forma, la respuesta retornará con un error. Es posible que existan errores en la transferncia y que la data requerida sea aún transmitida. Esto acontece por lo general cuando cualquiera de los otros campos retorna un problema (un número de programa malo, número de procedimiento malo, etc.). Los errores de autentificación siempre previenen al RPC de devolver la información requerida.

La identificación de un cliente es una parte importante de compartir archivos vía NFS. Hay disponibles muchos tipos de autentificación, debido al diseño abierto de RPC, permitiendo a desarrolladores de protocolo escribir nuevos y más poderosos métodos de autentificación. Entre esto se incluyen autentificación Diffie-Hellman (DH), autentificación Kerberos y seguridad RPC usando Generic Security Services API (RPCSec_GSS). Todos ellos incluyen un método ligeramente diferente de obtencion de autentificación, pero todos usan la misma idea básica. AUTH_SYS es el método de autentificación típico usado por máquinas GNU/Linux y UNiX. Esta basado en el credencial AUTH_UNIX usado por UNiX. Estas autentificaciones revisan el ID del usuario (User ID (UID)) asi como el ID del grupo (Group ID (GID)). Autentificar un usuario usando este método obliga al servidor a asumir que la información que el esta recibiendo es legítima. El propósito principal para esto no es la seguridad, pero el enrutamiento de la información y asegurarse de que el material esta disponible solo para el grupo especificado. El ingreso inicial del cliente es la línea de seguridad asumida. Esto deja grandes agujeros para acciones maliciosas, pero AUTH_SYS es usado típicamente para ambientes de pequeños negocios donde los usuarios son considerados merecedores de confianza. Si información sensible o usuarios desconocidos pueden accesar el servidor NFS, debe usarse un tipo de autentificación más segura.

Debido a la forma que UDP envia data en paquetes grandes completos, el RPC divide cadenas de dataen de una serie de archivos de longitud variable soportado por UDP. Este es el estándar de marcado de record. RPC organiza cadenas de dataen records que consisten en fragmentos de records más pequeños porque UDP no puede transportar paquetes más grandes de 64 KB. Esencialmente, el RPC simula una cadena continua de data con la organización y el envió de la data lo más rápido que sea posible en grandes paquetes. Este es el método del datagrama de enviar data. Un RPC o una respuesta debe caber dentro de un único datagrama y cualquier cosa más grande que un datagrama deberá ser roto en pedazos y enviado en partes. RPC reconoce el final de cada fragmento individual colocando una nota llamada último bit del fragmento al final del fragmento. Esto previene la mezcla de fragmentos de información corrompida. Cada medio de transporte tiene un MTU (Message Transfer Unit), la cual es la fragmentación básica de los datagramas.

PORTMAPPER

Portmapper administra conecciones RPC usadas por NFS y Network Information System (NIS). Los demonios que ofrecen servicios RPC le dicen a portmapper cual puerto ellos están escuchando. El servidor portmap debe estar ejecutándose en las máquinas que actuan como servidores para los protocolos que usan el mecanismo RPC. El servicio portmapper esta registrado en el puerto (well-known) TCP y UDP 111.

Los servidores RPC permiten al kernel elegir un puerto efímero, diferentes servidores normales que unen a puertos well-known. En otras palabras, el puerto RPC puede cambiar cada vez que el sistema es reiniciado ya que RPC utiliza cualquier puerto disponible. Los clientes, de este modo, deben contactar el portmapper para recibir el número del puerto efímero.

En un servidor RPC, una colección de procedimientos es llamado un programa y es identificado por un número de programa. Los servcios asociados con números de programas pueden ser encontrados en el archi-

vo /etc/rpc. Un ejemplo de un archivo /etc/rpc se muestra acontinuacion:

```
# ident" @(#) rpc  1.11  95/07/14  SMI"  /*  SVR4.0  1.2  */
# RPC
portmap      100000  sunrpc rpcbind
rstat        100001  rup perfmeter rstat_svc
rusersd      100002  rusers
nfs          100003  nfsprog
ypserv       100004  ypprog
mountd       100005  showmount
ypbind       100007
walld        100008  rwall shutdown
yppasswdd    100009  ypasswd
```

En el inicio, el servidor RPC se une a un número de puerto y le informa al portmapper de este número junto con los parámetros de los servicios, en la cual incluye el único número de programa. Todos los números de programas están mapiados a un puerto TCP o UDP específico. Los programas clientes contactan al portmapper para enterarse de cuales puertos un programa particular esta mapiando. Un cliente consulta al portmapper (en el host servidor de RFC) para obterner el número de puerto nesecitado para accesar a un servicio en específico. Una vez el cliente recibe el número de puerto, este contacta los números de puerto para consultar los servicios.

Los archivos /etc/hosts.allow y /etc/hosts.deny pueden ser usados para permitir o denegar a hosts el uso de portmapper. Las entradas en el archivo hosts.allow superceden las entradas en el archivo hosts.deny. Por ejemplo, la entrada "portmap:ALL" en el archivo /etc/hosts.deny negarán acceso a cada uno. Los hosts que requieren acceso al portmapper pueden entonces ser escritos en el archivo /etc/hosts.allow. En un sistema GNU/Linux, hay muchas máquinas que nesecitan accesar al portmapper. El portmapper administra nfsd, mountd, ypbmd, ypserv, pcnfsd y servicios como el ruptime y rusers, como se muestra en el ejemplo del archivo /etc/rpc. La entrada en el archivo /etc/hosts.allow, la cual daría acceso al host 12.126.186.70, como a continuación: **portmap: 12.74.23.87**

Si un administrador nesecita dar acceso a todos los hosts en esta sub-red, la entrada sería la siguiente:
portmap:12.74.23.0/255.255.255.0

Nombres de hosts o nombres de dominios no deben ser usados en el archivo /etc/hosts.allow.

Si el demonio portmapper falla o detiene su ejecución por errores no planificados, entonces todos los servidores RPC tienen que ser reiniciados porque RPC pierde toda la información de los puertos. Pero, portmap debe ser iniciado antes de cualquier servidor RPC.

Portmap puede ser iniciado con el siguiente comando:
$ /etc/rc.d/init.d/portrnap start

Y puede ser detenido con el siguiente comando:
$ /etc/rc.d/init.d/portmap stop

El portmapper más usado comúnmente en GNU/Linux es portmap. Rpcbind es otro nombre dado a versiones nuevas de portmap. Ambos protocolos usan el servicio RPC registrado en well-known puerto 111.

RPC esta quipado con algunas funciones depuradoras que ayudan a mostrar información de portmapper

y eliminar entradas de portmapper descontinuado. Para obtener toda la información acerca de todos los servicios RPC ejecutándose en un host en específico pueden ser encontrados usando rpcinfo.

En el ejemplo que sigue, el comando rcpinfo es usado para encontrar información acerca de todos los servicios RPC ejecutándose en el host miguel:

```
$/usr/sbin/rpcinfo   -p   miguel
program   vers   ptoto   port
100000    2      tcp     111    portmapper
100000    2      udp     111    portmapper
100007    2      udp     932    ypbind
100007    2      tcp     934    ypbind
```

Este comando da información acerca del número del programa, la versión del programa, protocolo, puerto y el nombre del programa. Esto puede ser de mucha ayuda para encontrar puertos ocupados (puertos que no han reconocidos la terminación del programa y continua esperando por información de el) por programas rotos. Ocasionalmente, cuando un programa es terminado anormalmente, este no se limpia el registro portmapper y sus entradas pueden perpetuarse. Revisar los puertos y eliminar las entradas muertas es fácilmente efectuado usando el comando rcpinfo -d.

NIS (NETWORK INFORMATION SERVICE)

NIS es un sistema de base de datos distribuida usada para proveer archivos de configuración comunes y administración centralizada a los usuarios en una Red de Area Local (LAN). En un entorno distribuido, es deseado conservar la transparncia de la red a usuarios. Para lograr esta meta, toda la información común importante tales como nombres de hosts, claves, grupos, etc., nesecitan ser consistente en toda la red. En una red sin NIS, lograr esta consistencia es difícil, porque los cambios hechos a estos archivos comunes deben ser propagados a todos los hosts. Este proceso de propagar cada cambio a los hosts no solo consume tiempo pero además afecta negativamente al rendimiento del sistema. Cuando se agrega un nuevo host a la red, hay una perdida de rendimiento en la actualizacion de todos los otros hosts a causa del nuevo host. Para manejar estas tareas administrativas comunes, NIS debe ser usado.

NIS resuelve el problema de manejar los archivos comunes de configuración a traves de la provisión de una base de datos distribuida que coloca copias de estos archivos comunes de configuración en un servidor central. Todos los hosts pertenecientes al NIS recuperan la información solo desde el servidor central. Si un nuevo host es agregado a la red, entonces NIS nesecita modificar un archivo de hosts único en un servidor central y propagar este cambio a los otros hosts en lugar de modificar los archivos de hosts de cada host individual en la red. NIS es mejor situada para los archivos que no son especificos a host. Ejemplos de estos archivos son /etc/passwd, /etc/hosts y /etc/group. Archivos dependientes o especificos de host como es el /etc/fstab no deben ser almacenados en el NIS.

Los siguientes tópicos serán cubiertos en esta sección:

- Modelo NIS
- Introducción al NIS
- Archivos controlados por NIS
- Demonios
- Comandos
- Configuración del NIS
- Observando la Información del NIS
- Mapas
- Monitoreo y Resolución de Problemas del NIS

- Múltiples Dominios
- NIS y DNS
- Seguridad de NIS

Modelo NIS

El NIS es construido sobre el modelo cliente/servidor. El servidor es un host que contiene los mapas NIS (archivos de información) y todos los otros hosts son clientes que requieren o peticionan la información de los mapas. NIS tiene dos tipos de servidores: el servidor master (maestro), el cual es el dueño de los mapas y un conjunto de servidores slaves (esclavos), los cuales obtienen los mapas desde el servidor maestro y manipulan las peticiones de los clientes. El servidor maestro es el único servidor que puede modificar los mapas; todos los otros servidores reciben copias distribuidas de los mapas modificados desde el servidor maestro.

El muy similar al DNS en ciertos aspectos. Por ejemplo, el esta basado en el concepto de dominios; cada dominio es servido por uno o más servidores que contienen la información para el dominio. Pero, NIS trabaja en base a broadcast para encontrar los servidores y por esto realmente no es conveniente para la resolución de nombres en redes WAN. El primer servidor que responda será el que NIS usará para dirigir sus petciciones.

Ya que no se requiere de ninguna autoridad para ser en un servidor NIS y el servidor puede suplir servicios y información sensible como autentificación de claves, asegurar un servidor NIS puede ser un problema. De toda forma, algunas medidas puden ser tomadas para sobreponer algunos de los problemas, tales como direccionar al cliente a un servidor o servidores en particular. Además, las últimas versiones de NIS, llamada NIS+ usan Secure RPC, el cual sobrepone algunos de los problemas de seguridad. Pero recuerde que no todos los problemas de seguridad han sido resueltos y usted debe probablemente implementar un protocolo más seguro tal como LDAP, el cual discutiremos en la siguiente sección.

Introducción a NIS

El NIS (previamente conocido como Yellow Pages) unifica la administración de información relacionada con la red. Una copia singular de ciertos archivos es mantenida y actualizada en un sólo host. Esta información se hace disponible para otros hosts que la usarán.

Para confiabilidad y división de carga, la información puede ser replicada atravez de varias máquinas.

NIS no esta limitado a la administración de ciertos tipos de informaciones. Cualquier archivo que pueda ser accesado en campos claves puede ser administrado por NIS.

Dominios

El NIS introduce el concepto de dominios. Un dominio es un conjunto de información compartida entre un grupo de máquinas. La noción de un dominio es introducida por razones administrativas. Las máquinas en un departamento o grupo de trabajo por ejemplo, pueden ser agrupados dentro de un dominio, para asi que todos los hosts dentro de ese grupo pueden ser administrados como uno sólo.

Tome en cuenta que los dominios NIS no son necesariamente relacionados a dominios DNS. NIS y DNS trabajan independientemente y una red en particular puede ejecutar NIS, DNS o ambos. Pero, ya que ambos fueron introducidos para hacer más fácil la administración de las redes, a menudo ellos son operados para que dado un dominio NIS y también sea un dominio DNS.

NIS usa la noción de maestros y esclavos. Cada dominio tiene un servidor maestro donde los cambios son efectuados a la información. Un dominio puede además tener servidores esclavos. Un servidor esclavo repli-

ca la base de datos de los servidores maestros y responde a los clientes peticiones de información. Un cliente no puede decifrar si un servidor es maestro o esclavo solo cuando hay una excepción que lo nesecite saber.

Clientes

Como ya hemos dicho, NIS esta basado en el modelo cliente/servidor. El cliente procesa información requerida desde una base de datos que es accesadas mediante un servidor. Múltiples servidores son proveídos por la tolerancia de falla y para la división de carga atraves de los host. Los procesos clientes determinan cuales servidores en un dominio usar atraves de un proceso de binding.

Un cliente recupera información desde NIS llamando una rutina de su librera. La rutina de la librería proteje al cliente de como la información es obtenida usando NIS o hasta de que se está usando NIS.

Mapas

La información suplida por los servidores es organizada en mapas. Cada entrada en dentro de un mapa consiste de una llave y una información asociada. Al requerir, los clientes especifican el dominio de su interes, el mapa y la llave. El servidor retorna la información asociada con esa llave.

Almacenados en un formato binario (dbm), los mapas son creados desde archivos de texto tales como /etc/passwd y /etc/hosts.

Porque la misma información a menudo nesecita ser recuperada usando diferentes llaves, un archivo de texto en particular puede generar más de un mapa. Considerar el archivo /etc/passwd. Nesecitamos poder buscar a los usuarios tanto por nombre de usuario o el ID del usuario. Por lo tanto, dos mapas son creados: passwd.byname y passwd.byuid.

Archivos Controlados por NIS

Los archivos de configuración pueden tener propósitos estrictamente locales, una mezcla de propósitos locales y de redes o propósitos a todo lo ancho de red.

Archivos locales y NIS

Ya sea que el archivo local sea consultado o que sea NIS que sea consultado depende en el mapa en particular o archivo que este en uso. Los archivos de configuración locales son o reemplazados o aon augumentados por NIS.

Archivos reemplazados

Los archivos remplazados no son consultados para nada. El mapa NIS correspondiente sobrepasa los archivos locales completamente. Aunque los los archivos locales no son consultados del todo, no los elimines por que pueden ser nesecitados. El archivo hosts, por ejemplo, puede ser necesario en el inicio del sistema antes de que NIS se este ejecutándo. Los siguientes archivos son el conjunto de archivos reemplazados:

- hosts
- networks
- protocols
- services

Archivos Augmentados

Los archivos augmentados son consultados primero. Si la entrada no es encontrada, el mapa NIS correspondiente será consultado. El archivo password es un buen ejemplo. Si la entrada requerida no es encuentra, entradas de un marker (marcada) extra dirigen la búsqueda a los mapas NIS. Estas entradas marcadas inician con el simobolo +. El archivo password en un cliente por ejemplo, tendrá una última entrada pare-

cida a esta "+:*:0:0:::". Lo siguiente son archivos de ejemplo encontrados en el conjunto augmentado:
• passwd
• group
• aliases

NIS centraliza cualquier archivo de red. Los archivos que difieren en cada host, claro está serán administrado en base host.

Demonios

NIS usa los demonios ypserv y ypbind.

ypserv

El demonio ypserv es usado para referirse a los mapas NIS locales. El sólo se ejecuta en el daemon ypserv del servidor y puede ser iniciado y puede ser iniciado como detenido por los siguientes comandos del demonio ypserv del SRC (System Resource Controller):

startsrc -s ypserv
stopsrc -s ypserv

El demonio ypserv efectuá las siguientes operaciones de búsqueda en un mapa especificado con un demonio especificado:

Match	Toma una llave como argumento, iguala llave con los valores en un mapa y retorna el valor igualado.
Get_first	Regresa el primer par de valores de llaves desde el mapa.
Get_next	Retorna el siguiente par de valores de llaves en el mapa.
Get all	Retorna el mapa NIS entero.
Get_order number	Retorna información acerca de un mapa en vez de una entrada del mapa.
Get_master_name	Retorna el nombre maestro en vez de entradas del mapas.

ypbind

El demonio ypbind es usado por un cliente NIS para recibir los servicios de un servidor NIS. Primero, el demonio emite (broadcast) una petición en la red NIS para un servidor. Una vez un servidor es contactado, el demonio ypbind toma la dirección de Internet del host, el número de puerto y almacena en el directorio /var/yp/binding usando el nombre de archivo de nombrededominio.version. Si el cliente decide accesar los mismos mapas en el futuro, este almacena direcciones que pueden ser usadas para futuras referencias. El demonio ypbind es iniciado y detenido por los siguientes comandos SRC:

startsrc -s ypbind
stopsrc -s ypbind

Comandos

Entre los comandos usados por NIS se incluyen lo siguiente:
• domainname
• makedbm
• ypcat
• ypinit
• yppassword
• ypwhich
• ypxfr

domainname

El comando domainname es usado para mostrar o estableceer el nombre de los dominios. Un dominio es un conjunto de mapas de NIS. Los administradores lo usan para establecer diferentes polticias para diferentes sistemas asi que los usuarios pueden accesar archivos pertenecientes a ellos.

Para establecer el nombre de dominio NIS de nombre abiertos, use el siguiente comando:
domainname abiertos

Para ver el nombre de dominio, solo se nesecita el siguiente comando:
domainname

makedbm

El comando makedbm es usado para crear mapas NIS. Con la ayuda de este comando, ambos mapas NIS, los de formato DBM asi como los no-DBM peden ser creados. El comando makedbm toma el parámetro InputFile y lo convierte en dos archivos de salida: OutputFile.pag y OutFile.dir. Cada línea en el archivo de entrada es convertida en un record simple de DBM (Data Base Manager).

El comando makedbm es comúnmente invocado desde /var/yp/Makefile para crear mapas NIS. En /var/yp/Makefile, todos los caracteres, hasta que aparezca el primer espacio o tabulado, son tomados como la llave mientras que el resto de la línea contiene el valor de la información. El el carácter backslash en el final de la línea es usado para indicar que la información continúa en la siguiente línea. El comando makedbm no trata el símbolo # como carácter de comentario asi los clientes NIS deben ser capaces de interpretarlo.

Una vez un mapa es generado, es necesario propagarlo a todos los servidores esclavos. Para hacer esto, se usa el siguiente comando:
makedbm -b

Para crear un archivo de formato no-DBM, se usa el siguiente comando:
makedbm -u
Al ejecutar el comando anterior, makedbm deshace un archivo DBM, imprimiendo un archivo DBM con solo una entrada por línea, con un espacio simple separando las llaves de los valores.

ypcat

El comando ypcat es usado para mostrar mapas NIS asi como tablas de traducción de nickname. Para mostrar simplemente el mapa passwd, use el siguiente comando:
ypcat passwd

Para mostrar el mapa de los hosts en el dominio juegos.abiertos.org, use el siguiente comando:
ypcat -d juegos.abiertos.org hosts

Para mostrar el mapa de la tabla de traducción de nickname, usa el siguiente comando:
ypcat -x

ypinit

El comando ypinit es usado para inicializar ambos el servidor maestro y el esclavo. Para inicializar el servidor maestro, se usa el siguiente comando:
ypinit -m

Durante la ejecución de este comando, ypinit construye el subdirectorio /var/yp o /etc/yp para el dominio actual por defecto. Si el dominio por defecto es escrito, entonces el comando ypinit contruye a /var/yp/writers. Después de crear el subdirectorio del dominio, ypinit construye y coloca todos los mapas en este directorio. Después de hacer esto, le pregunta al usuario por la lista de servidores esclavos. Esta lista es colocada en el mapa ypservers. El no requiere que los hosts mencionados en este mapa neseciten ejecutar NIS al mismo tiempo, pero ellos si nesecitan convertirse en servidores NIS antes de que se haga la primera modificación a los mapas.

Similarmente, para ajustar un servidor esclavo, el comando ypinit es usado. La sintaxis para inicializar un servidor esclavo es "ypinit -s MasterName".

ypinit s miguel

Durante la ejecución del comando anterior, ypinit transfiere los mapas desde el servidor maestro miguel al sistema actual. Para que la transferencia tome lugar, ambos: el maestro y el esclavo deben estar en la misma red IP. Si ellos están en redes diferentes, entonces el esclavo sería ajustado como un cliente y el comando ypset sería usado.

yppasswd

El comando yppasswd es usado para cambiar la clave para la red NIS. El cambio de clave puede ser sólo efectuado por el dueño de la clave y aquellos quienes son los usuarios root en el servidor NIS.

Para cambiar la clave NIS, usa el siguiente comando:

yppasswd

Al ejecutar este comando, el sistema le pide al usuario entrar su clave vieja. Luego de entrar la clave vieja correcta, el sistema pide al usuario entrar la nueva clave y de nuevo le pide entrar la clave nueva de confirmación. Si ambas claves nuevas son iguales, entonces la clave NIS es cambiada.

Para cambiar la información GECOS de un usuario cuyo UserName (nombre de usuario) es miguel, use el siguiente comando:

yppasswd -f bill

Al ejecutar el anterior comando, el sistema le pide la clave del usuario. Cuando el usuario ingresa la clave correcta, entonces puede proceder a cambiar su información GECOS.

ypwhich

El comando ypwhich es usado para identificar el servidor maestro o un proveedor de servicio NIS para un mapa en particular o un dominio o un host.

Para encontrar el nombre del servidor NIS para la máquina local, simplemente use el siguiente comando:

ypwhich

Para encontrar el nombre del servidor NIS para un dominio como oficina.abiertos.org, use el siguiente comando:

ypwhich -d oficina.abiertos.org

Para encontrar el nombre del servidor NIS para un mapa llamado equipos, usa el siguiente comando:

ypwhich -d equipos

ypxfr

El propósito principal de este comando es transferir un mapa NIS desde un servidor NIS a un host local. Para lograr hacer esto, el comando crea primero un mapa temporal en el directorio /var/yp/Dominio en el cliente. Entonces este mapa temporal es llenado con entradas del mapa recuperadas desde el servidor. Después de que el mapa se llena, los parámetros del mapa (como son orden, número y servidor) son pasados a este mapa temporario. Entonces este comando busca y elimina cualquier versión vieja del mapa que pueda estar presente en el cliente. Finalmente, el mapa temporario es movido al mapa real. Para ejecutar este comando, se puede usar la herramienta de interfaz de administración del sistema.

Para busca un mapa desde un host llamado ivelis en otro dominio llamado oficina.abiertos.org, use el siguiente comando:

ypxfr -d oficina -h ivelis

Configurando el NIS

Primero usted debe configurar un servidor NIS maestro.

Servidor maestro

Para ejecutar operaciones que envuelvan cambiar mapas NIS o ejecutar procesos de demonios NIS, nesecitarás privilegios de root (administrador del sistema).

1. Establecer el nombre de dominio NIS en el servidor:
   # domainname nombre
2. Cambiese al directorio /var/yp y ejecuta ypinit:
   # cd /var/yp
   # ypinit -m

   (ypinit lo más probable es que este en /usr/etc/yp o /usr/sbin, asi es que nesecitarás los nombres de ruta completos si no están en su ruta de búsqueda de comandos o PATH.

   ypinit crea el directorio /var/yp/nombre-dominio. Este entonces crea los mapas NIS desde los archivos ASCII del sistema con la invocación “make”.
3. ypinit le preguntará si debe terminar en errores no fatales. Si respondes que si (en este caso “yes”), ypinit saldrá con el primer problema encontrado, permitiendole reparar y reinvocar ypinit. Esto es útil la primera vez que lo ejecute.
4. ypinit pide una lista de otros hosts que deben ser convertidos a servidores NIS. Estos hosts no necesitan estar ejecutándose NIS aún, aunque ellos deben estar la primera vez que que los primeros mapas se actualicen, ypinit crea el mapa ypservers usando esta información.
5. Iniciar ypserv:
   # ypserv

   (ypserv debe estar lo más probablemente en /usr/etc o /usr/lib/netsvc/yp.)
6. El servidor maestro NIS ahora esta ejecutándose. Para habilitar los procesos clientes en el servidor host maestro usalo iniciando ypbind:
   # ypbind

Servidores Esclavos

Un servidor NIS debe ser ejecutado antes de que puedas configurar los servidores esclavos.

ypinit es también usado para inicializar servidores esclavos.

1. Establezca el nombre de dominio NIS en el host servidor esclavo:
   # ypdomainname nombre-dominio
2. Edite los archivos locales del cliente para que asi los procesos consultando ha estos archivos se refieren a los mapas NIS. Es decir, agregar entradas NIS marcadas (normalmente una línea que inicia con un +).
3. Establecer el host un cliente NIS. Si su ypinit tiene una opción -c, inicialize el cliente, entrando los nombres de los servidores NIS:
   ypinit -c
   Inicie el ypbind:
   # ypbind
4. Inicialize el servidor esclavo:
   # ypinit -s servidor-maestro
5. Inicie el ypbind:
   # ypbind

Al agregar un servidor esclavo a una red, actualize el mapa del ypservers maestro. La razón de esto es

que cuando los mapas son modificados en el servidor maestro y luego en los esclavos, los mapas son propagados solo para esos esclavos listados en ypservers.

Clientes

Para configurar un cliente NIS, realize los siguientes pasos:

1. Establezca el nombre del dominio NIS en el cliente:

   # ypdomainname nombre

2. Edite los archivos locales del cliente para que asi los procesos que consultan esos archivos se refieran a los mapas NIS. Es decir, agregar entradas con marcadores de NIS (por lo general una línea que se inicia con un +).
3. Inicialize el cliente:

   # ypinit -c

   Escribir los nombres de los servidores NIS en orden de preferencia, físicamente los más cercanos primero.

   Algunas versions de ypinit no soportan la opción -c, si este es el caso, inicie el binder del NIS usted mismo:

   # ypbind

Una vez el NIS este configurado, los archivos de inicio del sistema (archivos rc) deben ajustar el nombre del dominio y dar inicio al demonio ypbind.

Observando la Información del NIS

Al igual que los comandos cat y grep que son usados para mostrar y buscar archivos de texto locales, ypcat y ypmatch son usados para obtener información desde servidores NIS, ypcat muestra el mapa entero (“ypcat passwd.byuid”); ypmatch busca un mapa dado para una llave (“ypmatch miguel passwd.byname”).

Ambos ypcat y ypmatch tienen alias (nicknames) para los mapas más comunes:

Nickname	Map Name
aliases	mail.aliases
ethers	ethers.byname
group	group.byname
hosts	hosts.byname
passwd	passwd.byname
protocols	protocols.bynumber
services	services.bvname

Note que estos no son entendidos por NIS como un total pero solo por ypcat y ypmatch. Para mostrar el conjunto de nicknames, use ypcat -x o ypmatch -x.

Mapas

Como NIS por lo general maneja un número de maps, la herramienta ideal para coordinar las actualizaciones es make. El make reconstruye solo esos mapas que han cambiado de acuerdo a las descripciones especificadas en un archive make (makefile). El makefile define el procedimiento para generar mapas individuales desde sus respectivos archivos de textos.

El makefile NIS por defecto también propaga los mapas a los esclavos llamando a yppush.

Propagando Mapas a los Esclavos

Los mapas son normalmente transferidos usando yppush, el cual es invocado por make y puede ser invocado interactivamente. Trabaja así:

1. yppush usa el mapa ypservers para contactar cada ypserv esclavo, requeriendo que efectue una transferencia de mapa.

2. El ypserv esclavo invoca a ypxfer, diciendole cual mapa transferir y como contactar a yppush en el maestro (el cual espera por una respuesta).

3. Habiendo intentado transferir el mapa, ypxfr envia de vuelta la condicion del mensaje a yppush, el cual le informa los sucesos en la tranferencia.

4. yppush muestra un mensaje que indica como todo transcurrio.

Si un esclavo pierde la transferencia de un mapa (quizas porque se encontraba fuera de línea en el instante cuando se cargó el servidor), queremos asegurar que los mapas no se desactualicen. El comando ypxfr puede ser llamado directamente desde el esclavo (normalmente es llamado por ypserv). Invocando ypxfr usando cron, podemos hacer que las transferencias de mapa ser efectuén regularmente. Los scripts (ypxfr_1perday y ypxfr_1perhour) son creados en base a que tan frecuentemente esperamos que los mapas particulares cambien.

Algunas versiones de NIS proveen un demonio ypxfrd en el servidor maestro, el cual acelera las transferencias de los mapas.

Actualizaciones a la Claves de Mapas

Las actualizaciones a las claves de los mapas son tratadas en una manera diferente que otras actualizaciones de mapas. Primero, son los usuarios, y no el administrador, quienes deben ser capaces de actualizar partes del mapa para asi poder cambiar sus claves. Segundo, idealmente los mapas deben ser propagados inmediatamente para que las clave cambie en toda la red inmendiatamente.

Como un NIS es una base de datos de solo lectura, un mecanismo por separado maneja las actualizaciones de las claves.

El demonio yppasswdd reside en el servidor maestro para los mapas de claves. El responde a peticiones desde un programa cliente, yppasswd, para cambiar las claves de los usuarios. El comando yppasswd es usadoigual que el comando passwd pero contacta el yppasswd de los servidores NIS maestros en lugar de modificar un archivo local.

El yppasswdd hace el cambio luega actualiza los mapas passwd (ya sea directamente o via make).

Monitoreo y Resolución de Problemas del NIS

NIS viene con un paquete de herramientas para ayudar en el depuramiento de sus operaciones. Al consultar los hosts, las herramientas de depuramiento intentarian traducir el nombre de un host dado a una dirección usando NIS. Si NIS no está trabajando correctamente, la traducción del nombre puede ser que no trabaje. Si ese es el caso, deberá específicar el host por su dirección IP.

Desplegando los Bindings del Cliente

El comando ypwhich consulta un cliente (ypbind) para sus binding actual. Sin argumentos, el ypwhich consulta el host local: **# ypwhich host**

Forzando un Cliente a Hacer el Binding con un Servidor Específico

Use ypset para forzar ypbind a hacer el bind con un servidor en específico:

ypset server

Algunas versiones de ypbind no permiten esta acción por defecto.

Podemos querer forzar el binding de un cliente si ypbind esta tratando un broadcast para encontrar un servidor pero el servidor esta en otra red (o la red local no soporta broadcat) o si, por ejemplo, estamos poniendo a prueba un servidor particular.

Verificando la Transferencias de un Mapa

El comando yppoll consulta un servidor por el número de orden de un mapa específico:

yppoll [-h host] nombre-mapa

El número de orden de un mapa es su número de ID, el cual es asignado automáticamente y es cambiado cada vez que un mapa es actualizado. Deberá cuestionar el servidor para asegurarse que los servidores están usando el mapa más actualizado.

Múltiples Dominios

Algunas circunstancias pueden ameritar la configuración de múltiples dominios NIS. La idea aqui es que el servidor maestro en un subdominio es en realidad un servidor esclavo en el dominio completo. Múltiples dominios son complejos de configurar, administrar y están más allá de una introducción a NIS como es está.

NIS y DNS

En ambientes de redes de hoy es muy possible encontrarse usando NIS y DNS al mismo tiempo. Hay tres opciones que se le pueden presentar:

1. Ejecutar NIS sin DNS. Si DNS está ejecutándose, las rutinas de busquedda de host se dirigirán solo al NIS.
2. Para la búsqueda de nombres de hosts, se dirige a NIS primero; si la dirección del host no se encuentra, consultar al DNS.
3. Sólo usar DNS para la búsqueda de nombres de hosts; usar NIS sólo para consultas de otros tipos de información.

Las dos ultimas opciones son las más satisfactorias. Al eligir la última opción consolida todas las búsquedas de los host bajo un servicio de nombre.

Versiones modernas de UNiX permiten que la orden de búsqueda sea específica, con el uso de un archivo de configuración (normalmente /etc/nsswitch.conf o /etc/svc.conf).

El nombre de dominio NIS es por lo general basado en el nombre del dominio DNS:

abiertos.org	DNS
info.abiertos.org	NIS

El sistema cliente puede ser configurado para usar varias combinaciones de archivos locales, NIS y DNS para poder accesar la información correcta. En este caso, NIS puede ser usado en lugar de el archivo /etc/hosts, o puede elegir el orden de los servicios a usar usando un sistema llamado nsswitch. Si sus planes son de usar NIS en toda su organización, tendrás que familiarizarse con la interaccion de DNS y NIS.

Seguridad NIS

La seguridad nunca ha sido uno de los puntos de atracción de NIS. La intención de este fué, claro está, un mecanismo para propagar información sobre la red; la seguridad al parecer no fué de prioridad. Una forma de quebrar la seguridad de una máquina sería solo agregar una máquina a una red local y ejecu-

tar en ella ypserv. Cuando un cliente ypbind broadcast para un servidor, nuestro host falso responderá, sirviendo nuestro falso mapa de passwd. Ahora nosotros solo ingresamos en una máquina cliente que ha hecho el bind con nuestro servidor usando el nombre de usuario y la clave que tenemos hecha.

Todo esto asume que tenemos acceso a una red local y una que use broadcasts. Por su naturaleza y diseño, las redes locales permiten a nuevas máquinas sean fácilmente agregadas y removidas. Sin enbargo, aún con la seguridad del NIS mejorada al punto de no permitir nuestro ejemplo del servidor falso, si podemos tener acceso al cable de red por la consecuencia del NIS ser tan abierto podemos accesar a la máquina por otras vias.

LDAP

En esta sección, cubriremos el LDAP (Lightweight Directory Access Protocol). LDAP provee servicios de directorio usando una base de datos distribuida. En este contexto, un directorio es una base de datos de objetos y sus atributos, generalmente usado para mantener recursos de red tales como IDs de los usuarios, hosts e impresoras.

Un directorio contiene más información descriptiva acerca de los objetosque usan los atributos. Por lo general es más leído que lo que es modificado, asi que el directorio es optimizado para leer y buscar. Además, los directorios tienden a no tener ni la capacidad de hacer ni deshacer transacciones sofisticadas o de alto volumen que tienen las base de datos.

La información del directorio es almacenada en pares de atributo-valor. En otras palabras, toda la información en la base de datos debe ser asociada con un atributo. El atributo describe la información y el valor es la representación de la data misma.

La versión 2 del protocolo LDAP esta definida en los RFCs 1777, 1778 y 1779.

Los clientes LDAP someten consultas o informaciones actualizadas a un servidor LDAP. Si el servidor determina que el cliente tiene los derechos de accesos apropiados, el devuelve con una respuesta o refiere al cliente a otro servidor LDAP.

Comparando a un directorio con una base de datos relacional, un directorio generalmente cambia con poca frecuencia pero si tiene accesos frecuentes de lectura. Además, un directorio es organizado en un árbol jerárquico en lugar de filas.

Los siguientes tópicos son discutidos en esta sección:
- Usos de LDAP
- Replicación y Sincronización
- Estructura del Directorio
- OpenLDAP

Usos de LDAP

El Navegador Netscape puede ser configurado para usar servicios LDAP en dos maneras diferentes. Primero, puede ser usado como una libreta de direcciones. El directorio contiene toda la información que normalmente sería instalada localmente en el cliente. Esto le permite tener un servidor LDAP actuando como una libreta de direcciones compartida, la cual puede ser accesada desde cualquier lugar con acceso a red. La otra forma que Nestcape puede usar LDAP es para una característica llamada perfiles flotantes (roaming). El servidor LDAP mantiene toda la información de perfil del usuario, incluyendo historia, apuntes (bookmarks), páginas de inicio y preferencias. Esto trabaja bastante bien para los usuarios quienes trabajan en más de una máquina, asi permitiendo entonces tener el mismo entorno de trabajo donde quiera que ellos

están. Este tipo de administración centralizada de la información es muy ponderosa y se ha convertido más común en los ambientes de trabajos empresariales. Configurar un servidor LDAP para ser usado para consultas de clientes Nestcape requiere la modificación del esquema de sus directorios.

Con el uso PAM, usted puede configurar su sistema para usar LDAP para que le de servicio de autentificación en los ingresos o logins. Esto requiere instalar y configurar el módulo apropiado de PAM que es pam_ldap. Una de las gran ventajas es que puedes centralizar la base de datos de las claves de los usuarios en un servidor LDAP en vez de tener un gran número de archivos /etc/passwd que necesariamente deben ser modificados constantemente.

Existen muchas herramientas disponibles para asistir en la administración y manipulación de las bases de datos LDAP. Hay un módulo para linuxconf disponible que le permite configurar los parámetros de un servidores LDAP. El es llamado Idapconf. Todas las distribuciones de mayor embergadura, incluyen cierto tipo de interacción con servidores de autentificación LDAP y además contienen herramientas para instalar y configurar su propio servidor LDAP. Está disponible un cliente KDE llamado kldap, el cual tiene una interfaz gráfica muy buena. Para los que usan escritorios Gnome, el programa equivalente es llamado GQ, disponible de http://biot.com/gq. Si estas migrando un sistema de archivos de configuración estándar de UNiX hacia un sistema LDAP, puede encontrar las herramientas de migración para LDAP en http://www.padl.com/tools.html que le pueden ser muy útiles.

Replicación y Sincronización

LDAP fué diseñado como una base de datos distribuida y la información puede ser replicada entre muchos servidores. El diseño distribuido de LDAP lo hace escalable; usted puede iniciar con un servidor LDAP y luego expandirse a muchos servidores, cada uno conteniendo sólo una porción de la base de datos. Usted puede distribuir porciones individuales del árbol independientemente. Por ejemplo, podemos almacenar la información del departamento de ventas en un servidor dentro del departamento mismo y la información del departamento de TI en un servidor en su departamento.

La replicación provee redundancia y tolerancia a fallas. Esta además incrementa el rendimiento porque usted puede accesar al servidor más cercano que contiene la información.

La replicación también hace que el sistema sea escalable. Si descubre que su servidor de servicio de directorio no esta respondiendo a consultas lo bastantemente rápido debido a está sirviendo a muchos clientes, usted puede dividir la base de datos entre varios servidores.

La parte difícil de mantener una base de datos distribuida es mantener todos los servidores consistentes. LDAP usa un proceso llamado sincronización para pasar cambios en la base de datos desde un servidor a otro. A diferencia de NIS, los servidores LDAP solo pasan actualizaciones diferenciales (o sea sólo la parte que efectuó cambio y no el total) cuando ellos sincronizan, no la base de datos completa.

No hay un método estándar de replicación, asi que aún no es posible replicar entre servidores LDAP de diferentes vendedores.

Estructura del Directorio

Una base de datos LDAP contiene una jerarquía de entradas, a veces llamadas objetos. Cada entrada tiene una clase de objeto, que define que clase de información esa entra contiene. La información es dividida en atributos. Una clase de objeto define los atributos, algunos de los cuales son requeridos y el resto son opcionales. Si las clases de objetos existentes no suplen sus nesecidades, usted puede definir sus propias clases de objetos.

X.500

LDAP usa la estructura de directorio X.500. Si usted está familiarizado con el producto de Novell NDS o el Active Directory, LDAP es muy similar.

La especificacion OSI X.500 es un servicio de directorio completo y exhaustivo que incluye un modelo de información, un espacio de nombre y una arquitectura o framework de autentificación pero cuando se junta con su Protocolos de Acceso de Directorio (Directory Access Protocol), se torna muy pesado para ser implementado ampliamente.

LDAP simplifica muchas operaciones X.500 y puede ejecutarse sobre redes TCP/IP ya existentes. Esto hace a X.500 disponible para una gama más amplia y variada y LDAP puede ejecutarse nativamente apoyado por su propia bases de datos. El puede proveer libros de direcciones privado y publico basados en servidores como los servidores de directorio Netscape y four11 (http://www.four11.com). LDAP puede además integrar redes TCP/IP con servicios de directorios propetarios como Banyan StreeTalk y Novell NDS.

Hay dos tipos diferentes de objetos. Contenedores, conocidos también como objetos intermedios, son usados mayormente para organizar la estructura del árbol. Ellos pueden contener otros objetos pero además pueden contener atributos. Objetos Leaf (Hojas), estos objetos no pueden contener otros objetos. Ellos contienen la información actual que usted está administrando. Entre los tipos de contenedores comúnes se incluyen:

OU	Unidad Organizacional (Oganizational Unit)
O	Organización (Organization)
C	País (Country)
DC	Componente de Dominio DNS (DNS Domain Component)

El tipo de contenedor DC see añadió recientemente a la jerarquía de sistema de nombre X.500. Está permite que su árbol de directorio tenga una estructura que es lógicamente la misma que la jerarquía DNS. Por ejemplo, si su nombre de dominio DNS es linux.com, usted usaria dn=linux, dn=com.

Todos las entradas de objetos requiren el atributo objectClass. El es algo comparable con un meta-atributo porque el describe el tipo de objeto, el cual en torno describe los atributos del objeto.

Cada atributo tiene una definición describiendo el tipo del valor. Los siguientes tipos de atributos están disponibles:

bin	Información binaria
ces	cadena con mayúsculas y minúsculas exactas (las mayúsculas y minúsculas son significativas durante las comparaciones)
cis	cadena con mayúsculas y minúsculas ignoradas (las mayúsculas y minúsculas no son significativas durante las comparaciones)
tel	cadena de número de teléfono (como cis, pero en las comparaciones se ignoran los espacios en blanco y guiones “-”)
dn	“distinguished name” («nombre distintivo»)

Un nombre distintivo consiste del nombre específico de la entrada y los nombres de las entradas contenedoras sobre el en la jerarquía.

Una entrada puede tener más de una clase de objetos. La clase de objetos específica los atributos que el objeto debe tener y los atributos que son opcionales.

El esquema del servidor es la definición de todas las clases de objetos y cuales atributos tiene cada clase de objeto.

Los controles de acceso pueden ser aplicados a las ramas del árbol, entradas individuales o tipos de atributos. Ellos pueden ser otorgados a grupos o individuos.

Nombres Distintivos

Cada entrada en la base de datos puede ser unicamente identificada por un nombre distintivo, el cual describe su posición en la jerarquía. Un nombre distintivo inicia con un nombre común, el cual es el nombre de la entrada misma. El nombre distintivo es entonces seguido por los niveles menos especificos en el árbol. El siguiente es un ejemplo de un nombre común:

cn=abiertos, ou=IT, o=La Fundacion, c=DO

CN significa nombre común, OU es unidad de organizacional, O es para organización y C es para los paises. Note que ellos van desde lo más específico al menos específico, lo que es el contrario de las rutas de nombres de archivos. Los elementos son separados por comas y dos puntos. Los espacios son permitidos y no se diferencia entre mayúsculas y minúsculas. Los espacios son significantes al menos que rodeen las comas.

NOTA: Si estas familiarizado con los NDS de Novell, recuerde que LDAP usa comas o dos puntos para separar elementos en un nombre distintivos, mientras NDS usa puntos.

OpenLDAP

El servidor LDAP más común en GNU/Linux es OpenLDAP. Como puedes deducir de su nombre, el es un proyecto Open Source, disponible en www.openldap.org.

OpenLDAP tiene tres tipos de backend. En casi todas las circunstancias, usarás el backend LDBM. Además hay el backend PASSWD, el cual permite acceso a un archivo estrándar /etc/passwd y el backend SHELL, el cual usted puede usar para tener scripts de shell arbitrarios que proveen la información en el directorio.

La base de datos LDBM usa un identificador único, llamado el EID, para cada entrada en el directorio. Hay un archivo index principal llamado id2entry y otros archivos index para las otras llaves index.

El LDIF (Data Interchange Format) es usado cuando se importa o se exporta. En LDIF, como en LDAP, cada entrada es únicamente identificada por un nombre distintivo.

Slapd

El demonio principal en el paquete OpenLDAP es llamado slapd. Si la base de datos LDAP es accesada con poca frecuencia, ustred puede ejecutarla mediante el inetd. Para hacer esto deberá agregar una línea similar a la siguiente en el archivo /etc/inetd.conf:

ldap stream tcp nowait nobody /usr/sbin/tcpd /usr/sbin/slapd -I

El -I le dice a slapd que está ejecutando desde el inetd en vez de stand-alone. Usted necesitará asegurarse de que el archivo /etc/services contenga una entrada para Idap, el cual es normalmente el puerto 389. No olvide enviarle la señal al inetd de releer su archivo de configuración.

Normalmente, el demonio slapd es ejecutado en modo stand-alone desde los scripts de inicio. Esto le permite al demonio almacenar (cache) información entre los accesos de los clientes. Esto también facilita mantener la consistencia del index cuando hay múltiples clientes actualizando la base de datos al mismo tiempo.

Usted debe siempre apagar el demonio slapd limpiamente para que de esta manerar el pueda liberar sus memorias temporales (buffers) y actualizar los archivos index apropiadamente. Para apagarlo limpiamente envie al demonio la señal TERM. La forma fácil para hacer esto es con el siguiente comando:

killall -TERM slapd

Configuración

El OpenLDAP es configurado editando el archivo slapd.conf. Dependiendo en como instaló OpenLDAP, este puede encontrarse en el directorio /etc/openldap o /usr/local/etc/openldap. Además puede especificar cual archivo de configuración usar especificándole la opción -f cuando inicias el slapd. El archivo de configuración es un algo diferente que la mayor parte de archivos de configuración en que las líneas comenzando con espacio en blanco son consideradas una continuación de la línea previa.

La distribución OpenLDAP viene con un ejemplo de archivo de configuración. Usted puede aceptar los por defecto en la mayoria de los casos de las configuraciones. De cualquier manera, nesecitará configurar los parámetros de configuración del suffix y rootdn. Estas configuraciones deben ser algo silmilar a la siguiente, asumiendo que su dominio es abiertos.org:

database Idbm
suffix “dc=abiertos, dc=org”
rootdn “cn=Manager, dc=abiertos, dc=org”
rootpw

El ldbm específica que las siguientes configuraciones aplican por defecto a la base de datos del tipo LDBM. El suffix establece el nombre de la cima de lka jerarquía de su árbol. Todo en la base de datos será un descendiente de el objeto suffix. Usted puede usar cualquier contenedor de nombre distintivo.

El rootdn específica el super usuario, quien siempre puede leer y modificar el directorio. En realidad, esto trabajaría aunque el objeto no exista en la base de datos. Esto le permite popular la base de datos iniciando con una base de datos completamente vacia. Uste puede querer dajar el objeto rootdn completamente fuera de la base de datos por razones de seguridad, ya que es difícil adivinar el nombre de rootdn si este no se encuentra dentro del directorio. Usted puede también especificar en este una contraseña para el usuario root.

El archivo de configuración es también usado para definir el esquema. Normalmente, esto se hace incluyendo otros dos archivos de configuración: slapd.at.conf y slapd.oc.conf. El archivo slapd.at.conf es usado para definir la sintaxis de los atributos usados en el directorio y el archivo slapd.oc.conf es usado para definir las clases de objetos, incluyendo sus nombres, atributos requeridos y atributos permitidos.

slurpd

El OpenLDAP usa slurpd para replicar las bases de datos entre los demonios slapd. El demonio slurpd verifica el diario (log) que es generado por slapd. Debe asegurarse de que slapd está escribiendo al archivo log, especificando la directiva replogfile en el archivo slapd.conf. Cuando se escriben nuevas entradas al log (diario), slurpd envía la información a los servidores esclavos de slapd. El hace esto creando una conexión LDAP normal a el servidor esclavo y escribiendo la información en ese servidor. El demonio slurpd se conecta al servidor esclavo con el DN definido en la directiva replica en el archivo slapd.conf

Programas Cliente

El OpenLDAP viene con algunos programas clientes. Recientemente, Novell se ha afiliado el proyecto OpenLDAP y va a usar los programas clientes como interfaz a sus propios servicios NDS. Los programas que más seguro usará son ldapsearch y ldapadd.

Probablemente usará el ldapadd para comenzar a popular su base de datos, aunque usted puede usar otros clientes. También usted puede utilizar las herramientas de migración LDAP disponibles en http://www.padl.com. También existen otros programas cliente están disponibles.

Para crear una base de datos fuera de línea, use el utilitario ldif2ldbm. Primero, debe crear el archivo

LDIF que desea importar. Usted puede ingresar su información manualmente o generala usando un procesador de texto. Por ejemplo, si su departamento de VENTAS le provee una lista de usuarios y sus informaciones, usted puede usar un script de perl para transformar la información al formato LDIF.

Además puede actualizar una base de datos LDAP existente utilizando este método. Lo más seguro que usted efectuará esto para cambiar el esquema. Primero, exporte la base de datos en formato LDIF. Luego, manipule el archivo LDIF, agregando cualquier información necesaria. Una vez usted tenga actualizado el archivo LDIF, ejecute Idif2ldbm para re-importar la información y reiniciar slapd. La siguiente es la sintaxis:

ldif2ldbm -i archivo.ldif -f /ruta/a/slapdf.conf

Ejercicio 7-6: Configurar y Usar OpenLDAP

Las soluciones para este ejercicio están incluídas dentro del contenido.

1. Instale el paquete OpenLDAP. Si su distribución viene con una versión preempaquetada, úsela. Si este no es el caso tendrás que descargar el paquete con los fuentes, desempaquetarlo, compilarlo e instalarlo.

2. Debe crear un directorio /var/ldap para almacenar toda la información de directorio de LDAP:
   # mkdir /var/ldap

3. Edite el archivo de configuración por defecto slapd.conf, haciendo los siguientes cambios a la sección que empieza con "database ldbm". El archivo de configuración por lo general estará en el directorio /etc/openldap, pero la ruta puede variar dependiendo en como ha instalado el paquete.
   suffix "dc=codigolibre, dc=org"
   directory /var/ldap
   rootdn "cn=admin, dc=codigolibre, dc=org"
   rootpw mypassword

   Normalmente, usarias una clave encriptada, pero para este ejemplo usaremos una clave de texto plano.

4. Kill/Mata cualquier demonio slapd ejecutándose:
   killall -TERM ilapd

5. Inicia el demonio slapd:
   /usr/sbin/slapd

6. Edita el archivo /etc/ldap.conf, asegurándose que las siguientes dos entradas que permiten a las utilidades cliente LDAP conectarse a nuestro servidor LDAP:
   host 127.0.0.1
   base dc=codigolibre,dc=org

7. Debe crear un archivo LDIF para popular la base de datos. Llámelo ejemplo.ldif y editelo para que su contenido sea lo siguiente:
   dn: dc=codigolibre, dc=org
   objectclass: domain
   dc: codigolibre

   dn: cn=Miguel, dc=codigolibre, dc=org
   objectclass: person
   en: Miguel
   sn: Usuario
   mail : miguel@codigolibre.org

8. Use ldapadd para popular el directorio con el archivo LDIF que creo:

$ ldapadd -f ejemplo.ldif -D "cn=admin, dc=codigolibre, dc=org" -W

se le pedirá digitar la clave de root:

Entra la clave LDAP :

Escriba la clave que usted definió en el archivo de configuración mypassword.

9. Ahora que el directorio esta populado; ejecutaremos algunas busquedas en el. Note que no hay nada particularmente especial acerca del comando ldapadd que fué usado para popular el directorio; siempre y cuando usted tiene los derechos de acceso apropiados, usted puede agregar o modificar las entradas en cualquier momento usando los comandos ldapadd o ldapmodify. Las configuraciones por defecto son ajustadas con acceso de lectura para todos y lectura/escritura para el rootn que especifico en el archivo de configuración.

10. Busque una entrada con el nombre común de Miguel:

$ ldapsearch -b 'dc=codigolibre,dc=org' 'CN=miguel'
cn=Craig, dc=codigolibre, dc=org
objectclass=person
cn=Miguel
sn=Usuario
mail= miguel@codigolibre.org

Usamos la opción -b para especificar donde iniciar la búsqueda. Normalmente, el por defecto es tomado desde el archivo ldap.conf. Fijese que la comparativa CN no es caso sensitiva.

11. Busque todos los usuarios que su dirección de correo electrónico contengan el nombre Miguel:

$ ldapsearch 'mail=*miguel*'

La salida sería la misma que la del paso anterior. Note el uso de metacaracteres (wildcards).

12. Imprima la salida de la base de datos completa a la salida estandar (pantalla):

$ ldapseach -L "objectclass=*"

Aqui imprimimos el contenido de todos los objetos en el directorio. La salida es un formato LDIF, asi les que se debe parecer exactamente al archivo de entrada usado para crear la base de datos. Asi que, puede usar la salida de este comando para modificar la información y reimportarla.

RESUMEN

En este capítulo, cubrimos un gran variedad de los difrente servicios de rede. Entre algunos de los puntos más importante se incluyen:

- DHCP es usado para proveer direcciones IP y otras informaciones de configuraciones.
- Los Servicios FTP permiten acceso remoto a archivos en el servidor.
- Telnet provee acceso de línea de comando al sistema.
- Debes implementar SSH en lugar de telnet para mejor los niveles de seguridad.
- Squid provee servicios de cache y proxy.
- INN provee accesos a noticias de red, conectandose a ambos servidore de noticias y clientes.
- La sincronizaciónde tiempo puede ser habilitada usando NTP.
- NIS provee información de configuración sincronizada atravez de los hosts de una red.
- LDAP puede ser usado para centralizar e integrar autentificación y otras informaciones.

PREGUNTAS POST-EXAMEN

Las espuestas a estas preguntas están en el Apéndice A.

1. ¿Por qué DHCP normalmente no es usado para asignar direcciones a servidores?

2. ¿Cuál comando FTP deben los servidores FTP implementar para permitir acceso de clientes que están detras de un firewall?

3- ¿Como se inicia un servidor NIS maestro?

4. ¿Usted telnet en su cuenta ISP vía Internet. ¿Qué puede usted hacer para asegurarse que no le roben su contraseña?

5. De un ejemplo usando ldapsearch para localizar un objeto en una base de datos LDAP con el nombre común de Jose Paredes.

6. ¿Cuál protocolo usan los servidores Squid para comunicarse entre si?

7. ¿Cuál protocolo es usado para trasportar artículos de noticias Usenet?



APACHE: SERVIDOR WEB

TÓPICOS PRINCIPALES	No.

OBJETIVOS

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Operar y realizar la configuración básica de Apache.
- Comprender y estar en capacidad de implementar Apache.
- Comparar y contrastar los siguientes servidores web: Apache, Apache SSL, mod__perl, mod_php3, and mod_frontpage.
- Describa los elementos que componen apache y describa la imprementacion del programa.

PREGUNTA PRE-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Quienes conforman el Apache Group?

2. ¿Cuál es el comando para instalar Apache desde RPMs?

3. ¿Dónde podemos obtener la última version de Apache?

4. ¿Cuándo inicia Apache, cómo se comporta? ¿Puede manejar múltiples peticiones de clientes a la vez?

5. ¿Qué puerto usualmente escucha Apache para aceptar peticiones?

INTRODUCCION

Apache es el servidor Web más usado, cuenta con más sitios web en Internet que cualquier otro producto Apache fué desarrollado como parte de un esfuerzo de voluntarios colaboradores a traves del mundo. Su meta fué construir una robusta implementacion de un servidor HTTP.

El grupo de voluntarios que manejan el desarrollo del servidor Apache es conocido como Apache group. Apache esta bajo constante desarrollo por este grupo. Además de numerosos usuarios de todo elmundo aportando sus ideas, trozos de código y documentacion al Apache group para la inclusión en el proximo lanzamiento. Al igual que linux, este proyecto open source fué trabajado con detenimiento.

Apache obtuvo su nombre que consistio en una serie de parches de código existente. El nombre es un juego de palabras de la frase “a patchy server”. apache fué desarrollado basado en el código y las ideas encontradas en el más popular servidor HTTP de aquel tiempo, el NCSA httpd 1.3. Desde entonces, ha estado constantemente bajo desarrollo por el Apache Group, un grupo de voluntarios que se encuentran en diferentes partes del mundo y se comunican mediante correo electrónico y otras tecnologias Internet para crear los parches.

La primera version oficial (0.6.2) del servidor Apache fué en abril del 1995. Por coincidencia, el centro nacional de aplicaciones para supercomputadoras (NCSA) reinicio su desarrollo en ese periodo, y el equipo de desarrollo de la NCSA ingreso al Apache Group como miembros honorarios, asi que los 2 proyectos compartieron ideas e hicieron correciones. Apache 0.8.8 fué lanzado en diciembre de 1995 y Apache 1.0 fué lanzado en diciembre de 1995 a menos de 3 anos después de haberse formado el grupo, el servidor Apache sobrepaso al NCSAs httpd como el servidor número uno en Internet.

Arquitectura Modular

Al igual que el kernel de linux, Apache tiene una arquitectura modular. Una de las características que hace versatil a Apache y fácil de personalizar es el uso de módulos. Los módulos son programas autonomos que pueden ser agregados o removidos de acuerdo a la necesidad del webmaster. El módulo batch se relaciona a una función específica

La ventaja de la natural cooperación del Apache Open Source es que los usuarios pueden escribir su propio módulo si desean una funcionalidad adicional. Apache exige que usuarios someten sus módulos a la Apache Software Foundation, una vez es completado el módulo podría ser certificado y estar disponible para otros usuarios. Como es natural, los módulos nuevos son creados frecuentemente y los módulos viejos actualizados. Sin embargo, el uso de módulos tiene su parte negativa. Cada módulo adicional cargado en el servidor por el usuario disminuira el rendimiento del servidor. Para un rendimiento optimo deseado, el servidor debería tener incluido solo los módulos que necesitan para su funcionalidad. También, cada vez que una nueva version de apache es lanzada, estos módulos deben ser actualizados, recompilados y redistribuidos. Para una completa lista de los actuales módulos de la última version de apache, visite los módulos de referencia http://modules.apache.org.

Uno de los más importantes y poderosos módulos que han sido desarrollados para Apache es mod_perl. Este módulo integra un interprete de Perl dentro del servidor web. Perl es un lenguaje scripting relativamente simple y poderoso que ha tenido alcance y popularidad para su flexibilidad en el World Wide Web (WWW), administración de sistemas y algunas otras tareas. Con mod_perl instalado, un escritor esta en capacidad para programar entero los módulos de Apache en el lenguaje perl, incrementando la facilidad de crear nuevos módulos. mod_perl también incremento la ejecución de los programas CGI Common Gateway Interface que son scripts en perl by interpreting the scripts itself rather than spawning the systems that Perl interprets.

INSTALACION

Apache viene en la mayoria de las distribuciones linux. Puede también ser obtenido mediante paquetes rpm o en código fuente en http://httpd.apache.org o en numerosos sitios de Internet. La última version de Apache disponible a la hora de escribir este libro era la 2.0.53 .

Los siguientes tópicos serán discutidos en esta sección:
• Instalación desde rpms
• Iniciando el Servidor
• Deteniendo el Servidor

Instalando desde RPMs

Si no instalaste apache como parte de la instalación del sistema operativo, puedes instalarlo desde un archivo rpm con el siguiente comando:

rpm -i apache-1.3.19.1386. rpm

En Debian puedes intalar la última version de Apache usando apt-get :
apt-get -install apache

Puedes también instalar el RPMs desde utilidades como GnoRPM. El programa de instalación instalara los archivos de configuración, scripts de inicio y otra documentacion. Usualmente estos diferentes archivos van en ubicaciones específicas.

Iniciando el Servidor

Normalmente, Apache es iniciado automaticaticamente por los scripts de inicializacion del sistema. Para iniciarlo manualmente, utilice el comando httpd. Este comando ira al archivo httpd.conf y correra el servidor con las configuraciones especificadas.

Algunas de las opciones de la línea de comandos para iniciar el servidor son:

-C directiva	Procesa directivas antes de leer el archivo de configuración.
-c directiva	Procesa directivas después de leer el archivo de configuración
-d directorio	Un directorio raíz alternativo elimina el archivo de configuración
-f archivo	Específica la ubicación del archivo de configuración

Una vez el servidor es iniciado, numerosos procesos hijos nacen de el. Para ver estos procesos escriba:
ps aux | grep httpd

Deteniendo el Servidor

Para detener el servidor Apache, inicie como el usuario root y escribe:
/etc/init.d/httpd stop

Ejercicio 8-1: Instalación de Apache (Opcional)

El propósito de este ejercicio es utilizar los comandos básicos de GNU/Linux para instalar la versión de apache en la distribución de GNU/Linux que tengas. Este ejercicio asume que tienes una distribución basada en RPM. La solución a este ejercicio son suministrados en el Apéndice A al final de este manual.

1. Remueva algún paquete rpm de apache si esta instalado.

2. Instala el rpm de apache que vino con su distribución (o alguno actualizado descargado desde algun sitio de Internet).

3. Localiza los archivos de configuración de apache. Cuales son sus nombres ? Examina a traves de cada archivo para mirar su contenido.

4. Inicia el servidor Apache. (El programa de instalación ya ha iniciado el servidor por ti).

5. Utilice un explorador para asegurarse que el servidor esta trabajando.

6. Crea un cambio menor en la pagina inicial HTML que apache visualiza asi sabras si realmente es tu pagina. Examina la pagina de nuevo para asegurarte que el cambio ha pasado .

7. Utilice el comando ps para mirar que procesos están corriendo, incluyendo httpd

8. Para el servidor Apache.

CONFIGURACION

El formato de configuración preferido es colocado en un archivo, httpd.conf. El formato de configuración es grandemente utilizado y esta siendo lentamente desfazado, separado en 3 archivos: httpd.conf, srm.conf y access.conf. La version reciente de apache trabajara de ambas maneras y verifica los archivos de configuración cuando inician. Toda la configuración y el comportamiento de tu servidor web se almacena en los archivos de configuración. Es recomendable hacer una copia de estos archivos después de la instalación asi puedes recuperar tu sistema en caso de que se corrompan.

Puedes elegir la ubicacion donde quieres colocar estos archivos, pero generalmente se encuentran en /etc/httpd/conf o en /etc/httpd/apache.conf.

Una vez instalas el servidor apache y te aseguras de tener todos los archivos de configuración listos, puedes comenzar a utilizar tu servidor. Apache creara una pagina inicial automáticamente. Esa pagina por defecto estara creada en tu directorio /htdocs bajo los directorios de instalación de apache

Los siguientes tópicos son discutidos en esta sección:
- Estructura Básica del Archivo de Configuración
- httpd.conf
- srm.conf
- access, conf
- index.html

Estructura Básica del Archivo de Configuración

Sin importar el número de los archivos de configuración usados, todos tienen el mismo formato. El formato es: **directiva opción opción**

Estos archivos podrian tener secciones (pseudo HTML tags) que podrian verse como el siguiente:
<directive option>
sub-directive option option ...
sub-directive option option ...
</directive>

httpd.conf

httpd.conf es el archivo de configuración principal. Este archivo contiene la información básica del sistema requerida para correr el servidor apache. El archivo en si consite en comentarios y entradas de configuración. Algunas de las entradas más importantes son discutidas en esta sección.

ServerType

El valor de la directiva ServerType puede ser standalone o inetd. En modo standalone el demonio httpd escucha conexiones. En modo inetd, inetd escucha por conexiones e inicia un proceso cuando llega la petición.

Port Number

Por defecto, el servidor Web corre sobre el puerto 80. Por seguridad, el servidor Web puede ser configurado para escuchar otros puertos como 8008, 8080, o algun otro puerto mayor de 1024.

HostnameLookups

Si la entrada es ON, entonces loguea el nombre o la dirección cliente después de realizar un reverse DNS lookup. DNS realiza una búsqueda a las llamadas del sistema bloqueadas las cuales toman una cantidad considerable de tiempo. Para evitar esta friccion, debe ser colocada OFF.

Usuario y Grupo

Ejecutando el servidor web como root puede comprometer el sistema porque cada script que se ejecuta sobre el servidor tiene prvilegios de root. Si específica un usuario sin privilegios, el efecto de alguna vulnerabilidad que pudiera ser explotada se reduce. Una configuración básica es como sigue:

ServerRoot

Este parámetro indica la ubicacion del directorio base donde se encuentra la configuración, error y los archivos log que el servidor encuentra

ServerRoot /etc/httpd

ServerName

Este es el hostname devuelto a los cliente que solicitan peticiones. Este nombre debe ser un nombre valido de DNS.

ServerName server-name, compc-iny.com

Server Admin

Cuando el servidor genera un mensaje de error, la dirección de correo indicada por esta directiva es usualmente agregada al mensaje para indicar la persona a contactar.

Server Admin dirección@server.org

srm.conf

El archivo srm.conf establece la configuración general como el árbol documento raíz del servidor y la ejecución de reglas relacionadas a los scripts CGI. Las configuraciones más importantes son descritas en esta sección.

DocumentRoot

Esta directiva específica donde se encuentra el directorio raíz para los archvos html. En algunas distribuciones, por defecto esta:

/home/httpd/html

En otras distribuciones será:

/var/httpd/html

UserDir

En la configuración por defecto, cada usuario podría publicar documentos web desde su cuenta. El nombre del directorio que es utilizado como directorio individual de root es asignado con la directiva userdir.

UserDir public/html

Si el usuario posee el siguiente directorio:

/home/username/public/html

entonces la pagina por defecto puede ser accesado especificando lo siguiente:
URL http://servername/~username

Directorylndex
Esta directiva indica los archivos por defecto del site. Es especificado como:
Directorylndex index.html index.htm index.shtml

Si hacemos la petición a la URL http://nombre-servidor, el servidor buscara primero recuperar desde index.html, luego index.htm y finalmente index.shtml. Si no encuentra el archivo, el servidor retornara la lista de directorios o un mensaje de errror basado en otras opciones de configuración.

AccessfileName
Para asegurar directorios individuales con clave, un archivo de control de acceso podría ser colocado en algun directorio del documento web. El nombre de este directorio es especificado con esta directiva:
AccessFileName .htaccess

ScriptAlias
Esto da un alias al directorio conteniendo los scripts ejecutables del servidor. El formato es:
ScriptAlias Ruta_URL directorio
Un ejemplo es:
ScriptAlias /cgi-bin/ /home/httpd/cgi -bin/

Cualquier petición a un archivo .cgi o algun otro script ejecutable configurado en este directorio será ejecutado por el servidor. El directorio actual de scripts debería estar fuera de tus documentos web, y la ruta completa se debe estar dada en el directorio.

AddHandler y AddType
Estas dos directivas usualmente trabajan juntas. AddHandler permite archivos con una extension específica ser mapeados a una accion particular. AddType crea un MIME type para una extension específica. Los usos principales de estas directivas son permitir la ejecución de scripts CGI fuera del directorio de scripts CGI y ser interpretados por el servidor.
AddHandler cgi-script .cgi

La directiva previa indica que los archivos con extension .cgi encontrados fuera del directorio CGI deben ser ejecutados. Si no es especificado, el usuario vera el código cgi returnado. El proximo ejemplo muestra como la interpretacion HTML del servidor es habilitada.

access.conf
El archivo access.conf es utilizado para fijar el control de accesso al servidor y los directorios web. access.conf configura los derechos para habilitar el directorio HTML root y el directorio CGI. La configuración de estos directorios es mostrada en esta sección y los derechos de acceso para otros puede ser configurada de la misma manera.

```
HTML Directory
<Directory /var/httpd/htrnl>
Options [None, Includes, Indexes. FollowSymlinks, ExecCG! and MultiViews]
AllowOverride [None, All .Options, Filelnfo.AuthConfig, Limit]
order allow, deny
allow from all
</Di rectory>
```

La etiqueta de apertura específica el directorio que aplican todas las directivas. Las opciones de directiva indica los archivos en el directorio que se comportaran de cierta manera segun el valor dado. Ninguno se utiliza cuando el sitio web utiliza solo archivos HTML estaticos, AllowOverride indicates the overriding options for .htaccess over entries in the global access.conf file.

Las directivas de control allow y order dicen quien tiene acceso a las páginas en el directorio web.

Habilitando el Directorio CGI

En el siguiente código, la opción ExecCGI es habilitada si quieres ejecutar scripts CGI fuera del directorio especificado en ScriptAlias en srm.conf:

```
<Directory /var/httpd/cgi-bin>
AllowOverride None
Option None
</Directory>
```

index.html

El programa de instalación de apache generara una pagina de prueba llamada index.html la cual será vista cuando cargues tu explorador mientras tu servidor este trabajando apropiadamente. Este archivo esta escrito en HTML. Necesitas tener conocimientos HTML para poder modificarlo. A continuación un ejemplo de una pagina de prueba en HTML:

```
<HTML>
<HEAD>
<TITLE>Pagina de Prueba del Servidor WEB</ TITLE> </HEAD>
<!-- Background white, links blue (unvisited) , navy, visited) , red (active) -> <BODY BGCOLOR-"#FFFFFF"
TEXT-"#000000" LINK="#OOOOFF" VUNK ALirJK="//FFOOOO" >
<H1 ALIGN="CENTER">Funcionó!</Hl>
Si puedes ver esta pagina significa que el software<AHREF-" http://www.apache.org/" >Apache</A> en este sistema
GNU/Linux se instalo con exito. Ahora podrá agregar contenido en a este directorio y reemplazar esta pagina de prue-
ba.
</BODY> </HTML>
```

Ejercicio 8-2: Configuración de Apache

En este ejercicio, haremos cambios en tres archivos clave de configuración de apache: httpd.conf, srm.conf y access.conf. La solución a este ejercicio son proporcionados en Apéndice A.

1. Crea una copia a los archivos de configuración

2. Invierte algun tiempo revizando el contenido de estos archivos para que te familiarices con las diferentes directivas en cada archivo de configuración.

3. Deten el servidor apache. Cambia el usuario bajo el cual corre httpd, desde el default, reinicia apache. Puedes verificar que el cambio fué satisfactorio utilizando ps.

4. Proporciona a su servidor un nombre conveniente.

5. Cambia la directiva ServerRoot. Asegúrese que crea copias de todos los archivos de configuración necesarios.

6. Ahora Cambie también la directiva ErrorLog.

7. Crea un directorio nuevo con un index.html diferente y otro conteniendo el que tu quisieras crear. Esto es bueno para practicar HTML. Ahora cambia el DocumentRoot del servidor. Utilice su explorador para

verificar los cambios hechos. No olvide refrescar la pagina.

8. Asegúrese que posee diferentes cuentas de usuario en su sistema linux. Cree las páginas personales para estos usuarios y asegúrese que pueden explorarlas.

SEGURIDAD

Los siguientes tópicos son discutidos en esta sección:

- Directorios ServerRoot
- Scripts CGI
- .htaccess
- Directorio/ Control de Archivos
- Acceso Basado en Host
- Autenticacion de Usuarios y Grupos
- Secure Sockets Layer (SSL)
- Interrupcion de una sesion SSL
- Certificados
- OpenSSL
- Módulos Apache
- Rendimiento de Apache

Directorios ServerRoot

Cuando inicias tu servidor apache, este inicia como root. Necesitas ser cuidadoso con esto. Asegurate que todos los archivos y directorios esten protegidos de modificación por parte de usuarios normales. No olvides incluir los directorios.

Por ejemplo, si ServerRoot va a estar /usr/www/apache, Necesitas ejecutar los siguientes comandos:

```
mkdir /usr/www/apache
cd /usr/www/apache
mkdir bin conf logs
chown 0.    bin conf logs
chgrp 0.    bin conf logs
chmod 755.   bin conf logs
cp httpd /usr/www/apache/sbin
chown 0 /usr/www/apache/sbiri/tittpd
chown 0 /usr/www/apache/sbin/httpd
chmod 511 /usr/www/apache/sbin/httpd
```

Scripts CGI

Apache provee soporte completo para scripts CGI . Un mecanismo para que un usuario someta información a la Web es frecuentemente necesitado. Esta información podría ser tan simple como una dirección de correo, o el sitio web podría solocitar algo tan complejo e importante como un número de tarjeta de credito. En cualquiera de estos casos, los scripts CGI pueden ser utilizados. El CGI scripting provee un método para la petición de información a un sitio por parte de un usuario, recibir dicha información y ejecutar el programa en el servidor para procesar la información. Asi, los scripts CGI pueden ser utilizados para comunicarse con estos otros programas sobre el servidor.

CGI no es un lenguaje de programacion, no esta especificamente relacionado con algun lenguaje de programacion sencillo. Es una interface que permite al servidor web resolver tareas comunes en un número diferente de lenguajes de programacion. Un script CGI puede ser un programa interpretado o compilado. Mientras el archivo es un ejecutables Apache no se refiere al tipo de datos que contiene.

Apache tiene un directorio estandar para los scripts CGI localizado en /usr/local/apache/cgi-bin (en algunas distribuciones, esta en /usr/lib). Cuando un archivo desde este directorio es llamado en la pagina web, apache automáticamente mira si es un script CGI y lo ejecuta, Restringiendo todo el directorio cgi-bin, el webmaster aprieta la seguridad. Esta restriccion también permite al webmaster a monitorear más facilmente todos los scripts corriendo en tiempo real.

.htaccess

Apache puede utilizar archivos .htaccess en el sistema de ficheros local para configurar permisos de acceso. Estos pueden ser muy utilizados pero también la configuración del servidor resulta algo complicada, asi que lo mejor es prevenir utilizar esta característica.

Si usted tiene una necesidad para la flexibilidad que los archivos .htaccess proporcionan puedes agregar seguridad para los directorios especificos como se requiere. Tenga en mente que los permisos de los directorios son recursivos, necesitas configurar la seguridad en los niveles superiores y flojar la seguridad en aquellos directorios que realmente necesita:

```
<Directory  />
AllowOverride None
</Directory>
<Di rectory /space)
AllowOverride All
</Directory>
```

El ejemplo anterior deja seguridad firmemente sobre el directorio raíz y deshabilita el uso del archivo .htaccess pero abre el directorio /space para utilizar archivos .htaccess.

El uso extensivo de archivos de acceso local puede crear problemas de rendimiento. Asumiendo que el árbol de directorio fué configurado para utilizar archivos .htaccess, una petición para el archivo /www/sales/us/mo/summary.html debería leer archivos .htaccess en /www/sales/us/mo. Esto sucede por cada petición.

Directory/File Control

Por defecto, si el servidor puede encontrar una via al archivo A utilizando una URL,lo proporcionará al cliente. Asi es mejor bloquear todo y solo agregar aquellos que son necesarios. Puedes hacerlo como el ejemplo mostrado a continuación:

```
<Directory  />
Order denyTal1ow
Deny from all </Directory>
<Directory  /usr/www/users/*/public/html>
Order deny,allow
Allow  from all </Directory> <Directory  /www/apache/httpd>
Order deny,allow
Allow  from all </Directory>
```

Esta directiva negara el acceso al directorio raíz / a todo el mundo y entonces los directorios /usr/www/users/public/html y /www/apache/httpd son asignados privilegios de acceso a los clientes.

Acceso Basado en Host

Puedes controlar el acceso a tu servidor web basado en las peticiones por parte de los clientes a nivel de host dominio o dirección IP. Utilice la directiva allow y deny con order para controlar acceso por host.

-ru

3 usos comunes son:

- Permite todo excepto algunos:

order allow,deny
allow from all
deny from [host : IP]

- Incluye algunas y excluye el resto:

order deny,allow allow from [host \ IP] deny from all »

Incluye algunas y excluye solo ip específicas:

order mutual-failure
allow from [host \ IP]
deny from [_ host \ IP]

Cuando utiliza el control de acceso por host, se utiliza direcciones IP en lugar de nombres de host porque algunas personas pueden utilizar tecnicas de spoofing utilizando herramientas específicas, ejemplo.

allow from 192.168.2.7

Puedes también utilizar fragmentos de números IP en lugar de un número IP. Esto permitira o negara a cada cliente perteneciente a esa subred, ejemplo.

allow from 192.168.2

Autenticación de Usuario y Grupo

El acceso a los recursos del servidor web puede también ser restringido solicitando usuario y clave e incluso la membresia a un grupo. Los nombres de usuario, clave y grupos están usualmente almacenados en base de datos y accesados desde allí. Las bases de datos pueden estar en cualquier formato, desde archivos planos, a DBM, Oracle, MySQL, y Sybase.

Para utilizar nombres de usuario, claves y grupos, usted necesita generar la base de datos con la clave y el grupo y agregar las directivas de autorizacion a los archivos de configuración de Apache.

Las bases de datos más simples de claves y grupos son creados en archivos de texto plano, ubicados en algun lugar del sistema de archivos. es común colocarlos en el directorio de configuración de Apache. Puedes utilizar el comando htpasswd para generar el archivo de claves, agregar usuarios y configurar claves.

Los comandos siguientes crearan el archivo de base de datos password en /etc/httpd/conf, agregar el usuario testing, entonces te pregunta que entre y repita la clave para el usuario testing:

htpasswd -c /etc/httpd/conf/passwd testing
New password:
Re-type new password:
Adding password for user testing

Si miras el archivo una vez es creado, podras ver un formato similar al siguiente:

cat /etc/httpd/conf/passwd
testing:rgbWtCGDxrmZys

La base de datos del grupo es similar al siguiente archivo group:

executives: miguel antonio
supervisors: ivelis Jazmine

Una vez ha creado el archivo de la base de datos de la contraseña y el grupo, entonces podras establecer las directivas en su archivo de configuración de Apache. Entre las directivas se incluyen:

Authname name	Nombre a dar al espacio de autentificación que esta siendo protegido
AuthType type_name	El tipo de autentificación a usar (Basic, MD5). El archivo que contiene los nombres

	de usuarios y sus contraseñas.
AuthGroupFile file_name	Este archivo contiene los grupos y sus miembros.
Require conditions	Condición a cumplir para ser concedido acceso.

Ejemplos:

Require user miguel ivelis	Sólo los usuarios miguel y ivelis
Require group manager	Solo esos en el grupo manager
Require valid-user	Todos los usuarios en AuthUserFile

Secure Sockets Layer (SSL)

La información que pasa a traves del Internet de manera no encriptada podría ser facilmente interceptada mientras se enruta entre hosts. Para algun tipo de información, esto no es realmente preocupante y protocolos planos tales como HTTP son suficientes. Para otro tipo de información, tales como tarjeta de credito u otra información delicada, un canal de comunicación encriptada es correcto para prevenir paquetes indeseados de sniffing como aquellos que pasan subredes o routers.

En la Web

Netscape aparecio con la version inicial de SSL en 1994. Motivados por su diseno que incluyo el establecimiento de conexiones seguras y creando procesos tan transparentes como sea posible al usuario final. Por anos, estuvo envuelto en un protocolo maduro que soporta muchas cifras y es la fundacion para las buenas transacciones web hoy dia de e-commerce.

Mientras el enfoque de SSL tiende a ser el uso de transacciones seguras sobre HTTP, Mantenga en mente que trabajara con algunas aplicaciones que son disenadas para operar sobre las capas de transporte y sesion de TCP/IP.

Dividir la Sesion de SSL

Una sesion SSL tiene un número de cosas que ocurren para establecer una conexión. Algunas negociaciones necesitan ser llevadas entre el cliente y el servidor para asegurarse que ambos están hablando los mismos protocolos. Los acontecimientos principales que suceden para establecer una conexión son discutidos aqui. Esta secuencia es también llamada handshake.

The Handshake

El SSL handshake contiene muchos componentes, cada uno utilizado para el establecimiento de una conexión segura. Al igual que la conexión de 3 vias TCP/IP que inicia una conexión simple TCP, El SSL handshake establece una conexión segura entre un cliente (browser) y un servidor Web.

Seleccion de Cifra

El cliente y servidor negocian el tipo de cifra a ser utilizada. Al final el resultado podría ser RC4 con un 40- o 128-bit o alguna de las variantes DES. Esos que fallan, ahi están los cifrados IDEA y Fortezza. Ambos, el mensaje Hello del cliente y el mensaje Hello del servidor listan los cifrados disponibles para la conexión. En este punto algunas otras cosas son determinadas, tales como las llaves que son intercambiadas y uno de los tres tipos de mensajes resumidos.

Intercambio de Llave

Habiendo determinado el método para el intercambio de llaves en el paso anterior, el intercambio es realizado. En el mundo de la criptografia, existen métodos para asegurarse pasar una llave entre 2 computadores a traves de una conexión insegura como lo es Internet. OpenSSL por defecto utiliza con seguridad Diffie-Hellman.

Conexión

Ambos lados ahora tienen una cifra común para usar y también tienen las llaves necesarias para encriptar y desencriptar la data que pasa entre ellos. Además de ser encryptado, la información podría ser manipulada de otras maneras para protegerlos, dependiendo que cifra esta siendo usada.

Los navegadores web que implementan SSL, usualmente tendran un idicador que muestra si una conexión segura ha sido establecida. En Netscape. es representado por un icono en la parte izquierda baja del browser. Cuando es cerrado, se ha establecido una conexión segura.

Certificados

Un aspecto importante de SSL implica el tener de cierto método creíble para determinarse quién está en el otro extremo de la conexión. La respuesta del SSL a esto es un sistema de los certificados que contienen la información sobre un sitio así como identificar llaves. Adicionalmente, los terceros pueden estar implicados que pueden atestiguar para la identidad de un sostenedor del certificado. Estos terceros también se refieren como autoridades del certificado.

Autoridades de Certificado

Con SSL, un site puede crear su propio certificado y entonces lo utiliza como su prueba que son quien dice que son, Obviamente, este tipo de certificado se convierte en más de una formalidad puesto que otros métodos tendrían que ser utilizados para verificar realmente quiénes es el sostenedor.

En una red interna dentro de una compañía, eso se podía verificar un número de veces, tales como dirección de red de una máquina o realmente pagando una visita al cubículo de esa persona.

Ya fuera en Internet, no es fácil establecer que alguien sea honesto sobre su identidad. Aquí es donde los terceros o un Certificate Authority (CA) se pueden utilizar para atestiguar para la validez alguien el certificado (identidad). Las funciones principales de CAs implican el publicar, verificar, peticiones del certificado de la firma. Actualmente, los CAs principales en el Internetson Thawte y VeriSign.

Generando Certificados

Como hemos mencionado anteriormente, no siempre es necesario utilizar un CA para proveer un sitio con certificado. La utilidad del openssl tiene una amplia gama de funciones incluyendo la capacidad de generar y de firmar certificados. Más de cómo generar su propio certificado se discute después.

OpenSSL

El comando openssl es una herramienta multiproposito para entender tareas de administración SSL. Existe una gran número de cosas que se pueden hacer con el comando openssl. Nos enfocaremos en una parte en que se basan los certificados.

La esfera entera de la funcionalidad del SSL se puede tener prácticamente con opciones que varían y las discusiones al openssl ordenan. Típicamente, se invoca de una manera similar a esto:

openssl function function-arguments Common

Funciones

La utilidad openssl es lo suficientemente larga que no podrias encontrar todas sus funciones, pero hay algunos que se necesitan con frecuencia. Esta sección que nos centraremos en alguno que se ocupe de la creación y de la firma de certificados.

req

Esta función esta centralizada en la generacion de certificados. La función req ofrece muchas alternativas dentro de el para personalizar un certificado. Un uso simple de esta función puede lucir algo asi:

openssl req -new out cert.csr -keyout key.pern

Invocando openssl con estos parámetros generara una llave 1024-bit RSA y entonces solicita que ingreses un password para ello. Una vez confirmaste el password, usted puede comenzar a incorporar la información para el certificado en sí, tal como el nombre de la compañía, la localización, etc. Una vez que esto finaliza, usted tendrá un certificado cert,csr y una llave para ese, key.pem.

rsa

Ahora usted tiene una petición del certificado y la afina.La llave necesita ser procesada más lejos de modo que sea usable por el web server y también los browsers que visitan el sitio. Procesando una llave a traves de la función rsa lo convertira en un formato aceptable. Otras opciones están disponibles para encriptar la llave en formatos alternativos asi como cosas que puede afinar la llave a los browsers visitantes. Aqui tenemos un simple uso de la función rsa:

openssl rsa -in key.pern -out server.key

Invocando openssl con las opciones de arriba generara una llave modificada llamada server.key, una vez usted ha proporcionado la clave ha creado la petición para el certificado.

x509

La función rsa maneja llaves; la función x509 maneja certificados. Esta función provee capacidades para manejar certificados incluyendo peticiones del certificado de la firma, certificados que expiran, y un anfitrión de otras características.

El paso final en la creación del certificado es convertir la petición del mismo creada con el req en un certificado real. El siguiente ejemplo toman los archivos que fueron previamente generados y los ejecuta a traves de la función x509. Las opciones proporcionadas dicen que la función x509 firma la petición y la certifica por 60 dias.

openssl x509 -in cert.csr -out server.crt -req -signkey server.key -days 60

Invocando openssl con esos argumentos procesara el certifica y enviara la salida a server.crt..

Después de completar estos pasos, tendrá una llave y un certificado que podra ser utilizado en el servidor web. Estos son, por supuesto, ejemplos basicos, existen muchas otras opciones que están disponibles, las cuales usted puede explorar revisando las páginas del manual:

- man req
- man x509
- man openssl

Ejercicio 8-3: Crear un Certificado con OpenSSL

Este ejercicio ilustrará el proceso de la creación del certificado contorneado previamente y permitirá que usted considere lo que parecería el certificado una vez en uso en un servidor web. No hay soluciones proporcionadas para este ejercicio..

1. Ingresa como root.

2. Crear un directorio temporal asi tendrá un lugar para trabajar con los archivos SSL.

   mkdir /ssltemp

3. Invoca el comando openssl con la función req para crear una certificado de petición.

   openssl req -new -out cert.csr - keyout key.pern

Ingrese y verifique una clave cuando le sea solicitada. Memoricela ya que la necesitara para el resto del ejercicio.

4. Después de ingresar y verificar su clave, te preguntará por la información que esta colocada en el certificado. Sientase libre a ser creativo con estos campos. Cuando es solicitado para introducir una clave, solo presione ENTER. Si deseas, puedes introducir un nombre de compania.

5. Invoca openssl con la función rsa.
openssl rsa -in key.pem -out server.key

6. Introduzca la clave creada en el paso 3.

7- Invoca openssl con la función x509:

openssl x509 in cert.csr -out server.crt -req -signkey server.key -days 60 Ahora, incorporaremos el certificado nuevo y la llave dentro del sistema por defecto en el servidor web.,

8. Copia el certificado dentro del directorio con que cuenta mod_ssl.
cp -f .server.crt. /etc/httpd/conf /ssl . crt 9-

Copie la llave dentro del directorio apropiado:

10. Apache necesita ser reiniciado para que los cambios tengan efecto. Debe hacerlo invocando el siguiente comando:
/etc/init.d/httpd restart

11. Si no esta corriendo el X, inicielo con el comando startx.

12. Cargue Mozilla desdel menu.

13. Escriba la siguiente dirección:
https: //local host

14. Observe que la información del certificado aparece, entonces haga click en Next.

15. Haga click en el boton More info para ver la información que fué introducida cuando estabas creando el certificado. Haga click en Ok para regresar a la pantalla del certificado.

16. Click en Next, Continue y Ok a traves de las páginas que siguen hasta que cargue finalmente. Note la posicion del padlock's clasp en la esquina izquierda más baja de la pantalla.

Módulos Apache

Una de las características de apache que hace que este servidor web sea extremadamente flexible y estable es el uso de módulos. La estructura modular de apache, como el kernel de Linux, permite al administrador web agregar y remover dinamicamente características para probar, templar por razones de seguridad. Los módulos más extensamente usados en Apache son para habilitar el servidor para la ejecución de scripts de Perl y la interpretacion de etiquetas embedidas con PHP o FrontPage para ampliar las capacidades dinámicas de HTML del servidor.

mod_perl

El módulo mod_perl Apache/Perl nos brinda el poder del lenguaje de programacion Perl y el servidor HTTP Apache, haciendo posible escribir completamente módulos de Apache en Perl. En adicion, el interprete de perl incrustado en el servidor evita la carga agregada de tener que encender a un intérprete exter-

no de Perl. La razon de mod_perl es permitir a usuarios hacer cosas con Apache que son más dificiles o imposibles utilizando expresiones regulares de CGI. Las ventajas principales de mod_perl son poder, velocidad y procesamiento personalizados. Una pregunta común de mod_perl es cuanto nivel de rendimiento da. Los desarrolladores han reportado velocidades que van de 200 a 2000 por ciento. El paquete mod_perl esta disponible en el sitio web de Apache http://perl.apache.org/dist/.

mod_php3

El PHP Hypertext Protocol (PHP) es un lado-servidor, un lenguaje scripting incrustado que permite peticiones a bases de datos a traves del servidor web Apache porque los scripts PHP son actualmente ejecutados en el servidor. PHP esta disenado para reemplazar los Active Server Pages (ASP) y es mucho más flexible que ASP, capaz de hacer cualquier cosa que otros programas de cgi pueden hacer, por ejemplo recolectar datos, generar contenido dinámico, y enviar y recibir las cookies. PHP actualmente soporta una variedad de bases de datos para plataformas Linux/UNIX pero también puede ser configurado para soportar Bases de datos Microsoft tales como Access y SQL Server.

mod frontpage

El paquete mod_frontpage permite extensiones de servidor frontpage para ser utilizadas en servidores Apache bajo plataformas Linux y Unix Y crea 2 modificaciones al software del servidor Apache. El módulo FrontPage para Apache intercepta las peticiones remotas desde los programas clientes FrontPage, validad seguridad y redirecciona la petición para el fpexe suid root stub. El programa fpexe acepta ser autor de peticiones desde el módulo Frontpage de apache, realiza una validacion extra seguridad, cambia la identificación del usuario al dueño del Web site que es sido autor, e invoca el apropiado servidor Frontpage para extensiones CGI ejecutables. El uso de mod_frontpage es opcional. Las extensiones del servidor frontpage serán funcionales en el entorno apache sin el parcho, pero frontpage explorer, no estara disponible para crear subwebs.

Agregando Módulos a Apache

La arquitectura modular de apache permite agregar nuevas funciones y flexibilidad al servidor de Apache. La mayoría del código que está en la distribución de Apache del propietario está en la forma de módulos, haciéndola fácil quitar y substituir los módulos según lo necesitado. Es fácil agregar módulos a Apache. Primero, el código fuente del módulo se debe obtener y poner en el directorio src de Apache. Entonces, la definición del módulo necesita ser agregado a la configuración de Apache. Siguiente, Apache debería ser recompilado y el servidor ejecutable debería ser reinstalado. Por ultimo, el servidor debería ser reiniciado. Una descripción más detallada sobre agregar un módulo a Apache puede ser encontrada en http://www.apacheweek.com/features/modulesoup.

Ajuste del Rendimiento de Apache

Un administrador de una red corporativa tiene caja de Linux con un servidor web configurado y de funcionamiento de Apache. Este servidor web, en promedio, sirve páginas web a una pequena poblacion. Una nueva estrategia de mercado trae un; explosión de tráfico al sitio web de la empresa dentro en las siguientes semanas. Todos en la empresa saben que la configuración actual no puede manejar el tráfico que espera.

Existen algunas estipulaciones a seguir. La principal y más importante, es que la instalación y configuración actual del equipo deben ser utilizadas. la única opción que nos deja es pellizcar los recursos de apache utilizando los archivos y parámetros de configuración proporcionados. Esta sección pasa a traves de dos cambios de configuración y luego da los pros y contras de estos métodos de preparación.

Método 1

El primer método para la optimizacion del funcionamiento del servidor es la modificación de todos los procesos hijos de httpd corriendo en cualquier momento. La metodologia de apache es tal, que cada pagina web servida utiliza un proceso httpd, muchos procesos que se ejecutan ocasionan que más páginas puedan ser ejecutadas en cualquier momento. Modifique la directiva StartServers dentro del archivo de configuración de Apache. La correcta configuración del archivo a veces depende de la distribución donde apache este instalado, suponemos que debería estar en srm.conf o httpd.conf dentro del directorio de configuración de Apache. La directiva StartServers controla el número de procesos httpd que deberian ser iniciados. Normalmente el número de procesos fluctua entre minSpareServers y MaxSpareServers dependiendo de la carga. Por defecto, Apache inicia 5 procesos inicialmente el cual puede fluctuar a diez (MaxSpareServers) Si el nivel de carga y petición lo requiere. Esta sección del archivo de configuración definen estas directivas, bajo una configuración por defecto:

MinSpareServers 5 MaxSpareServers 10
Número de Number of servers to start, initially - - debe estar en números razonables.
#
StartServers 5

Apache se adapta dinamicamente a la carga que ve, eso es, intenta mantener suficientes procesos del sevidor para manejar la carga actual, más algunos servidores para para manejar los puntos transitorios de la carga (e. peticiones simultaneas de un mismo explorador). El realiza esto periodicamente verificando cuantos servidores están esperando petición. Si hay menos Minispareservers, el crea nuevos, si hay más MaxSpareServers, algunos los apaga. Los valores por defecto en httpd.conf-dist son probablemente ok para muchos sites.

Utilizando el comando: ps -ax | grep httpd muestra la información sobre todos los procesos httpd que están actualmente corriendo en el servidor Apache por defecto mostrando la salida siguiente:

ps -ax | grep httpd
SALIDA

La salida muestra el servidor principal httpd y 5 procesos hijos esperando peticiones como están definidas en la directiva StartServers.

Porque un alto número de peticiones son esperadas, habra una serie de cambios realizados a la configuración por defecto. Lo más importante, doble la directiva StartServers e incrementa el MaxSpareServers de 10 a 15.

Ahora, después de reiniciar Apache con la configuración nueva, el comando ps -ax |grep http puede ser utilizado para asegurarse que la directiva era lo que estaba esperando. La salida debería ser como la siguiente:

ps -ax | grep httpd
SALIDA

```
0:00   (httpd)
0:00   (httpd)
0:00   (httpd)
0:00   (httpd)
```

Método 2

Otro método para la optimizacion del servidor es iniciarlo a traves de inetd. Si Apache es iniciado como un servicio de inetd, más bien como un demonio independiente, resulta en la habilidad de asegurar el servidor utilizando tcp wrappers, pero conservando recursos. Con la inicializacion de Apache bajo

inetd, solamente inicias el servidor cuando sea necesario. Esto es más lento y no es recomendado por el Apache Group pero puede tener sus ventajas. Primero, detenga el servidor, Siguiente, edite el archivo httpd.conf y cambie la directiva ServerType de Standalone a inted.

ServerType inetd

Esto le dice a Apache que correra bajo inetd. También le dice al demonio inetd que puerto escuchar y que comando utilizar para iniciar el servidor. Para hacer esto, edite /etc/inetd.conf y /etc/services. Si no existe agregue una línea en /etc/services que será como sigue:

www 80/tcp http # WorldWideWeb HTTP

Después que /etc/services es modificado, agregue la siguiente línea en /etc/inetd.conf:

AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Esta línea debe ser modificada de modo que el usuario (root en nuestro ejemplo) refleja bajo cual nombre de usuario el servidor web va a ejecutarse. También modifique el comando de inicio de Apache (/usr/sbin/httpd -f /etc/httpd/conf/httpd.conf en este ejemplo). Con estos 2 archivos modificados, localice el número de proceso utilizando el comando ps y use kill -HUP para reiniciar inetd. Después de usar este comando, intente el comando ps -ax |grep httpd. Esta vez, no debería haber ningún proceso httpd a menos que alguien este haciendo petición a este site.

Comparaciones de Cada Método

Mientras cualquiera de estos métodos es una solución valida, cada uno tiene sus propósitos. El primer método trabajaria mejor en situaciones donde existan recursos ilimitados y un mayor volumen de tráfico. Este método consume los recursos pero deja los procesos esperando en background en vez de esperar por ellos. Esta desventaja para esta solución es la consumicion mayor del recurso. Si el sistema de recursos esta disponible, esta es la mejor solución. Si el sistema de recursos no esta realmente disponible, iniciando Apache atraves de inetd es probablemente la mejor eleccion porque los procesos de Apache son solamente iniciados cuando hay un requerimiento de información. Este método también tiene sus problemas, el más prominente es referido a nosotros como Thundering Herd problem. Cuando una petición httpd es requerida inetd inicia un sin número de procesos httpd para tratar de controlar estas peticiones. Cada proceso utiliza algun tiempo de la CPU , pero solamente uno de ellos obtiene la petición y el resto muere hasta que la proxima petición llega. Otra razon de elegir el método dado es el segundo, además de la conservacion de recursos, es seguridad. Iniciando Apache a traves de inetd le agrega la habilidad para aplicar TCP wrappers. Aunque Apache tiene incorporada una utilidad para el control de acceso (principalmente a traves de archivos .htaccess), TCP wrappers puede ser utilizado, dejando las configuraciones de seguridad en un lugar para fácil administración. Así pues, si conserva los recursos mientras todavía acepta tráfico razonable es la preocupación principal, el segundo método podría ser la mejor opción.

Con cualquiera de los métodos indicados, la supervisión de recursos sigue siendo necesaria. Como administrador, hay muchos programas disponibles para la supervisión de los recursos del programa. Los dos más importantes en esta situación probablemente serán los comandos top y vmstat, top como es mencionado es su pagina de manual, proporciona una vista en curso de la actividad del procesador en tiempo real. Utilizando top muestra cuales procesos están consumiendo mayor memoria, la cpu y otros hechos relevantes. Si un proceso de apache llega a tener el 99% de uso de la CPU, mate el proceso puesto que parece tener algunos problemas. Utilizar vmstat puede ser dificil. La pagina del manual puede describir cada una de las columnas de salida cuando el programa funciona detalladamente. Para esta discusión, solamente si, so, bi, y bo serán monitoreados Estos están en memoria intercambiada del disco. bloques enviados al dispositivos y bloques recibidos desde un dispositivo respectivamente. Como un administrador, mira estos campos y posiblemente los utilizados en memoria (buff) y memoria en cache (cache). Si la memoria en cache esta en 0 mientras el campo

si esta alto, entonces podrian haber demasiados procesos hijo de httpd (o aplicaciones) corriendo en el sistema. Esto sería un problema real para el administrator y podría ser corregido modificando el número de procesos de apache iniciados o asegurandose que solamente los programas críticos del sistema esten corriendo.

Ejercicio 8-4: Seguridad de Apache

Las soluciones a este ejercicio son proporcionados en el Apéndice B.

1. Utilice el acceso basado en el host para prevenir que el computador accese a las páginas man localizadas en tu servidor mientras permite a otros continuar a tener acceso. Esto requerira más de un computador en la red.

2. Utlice la autenticacion de usuario para restringir el acceso al directorio de páginas man en su servidor que creo en el paso anterior. El acceso debería ser solamente concedido al usuario test1.

3. Cree un archivo de autenticacion de usuario en su directorio configuración apache.

4. Detenga y reinicie el servidor.

5. Utilizando su navegador, asegúrese que la seguridad esta trabajando. Pruebe ambos, su configuración de autenticacion de usuario y su configuración acceso basado en host.

6. Visite www.apache-ssl.org y revise sus opciones para utilizar SSL sobre Apache.

7- Lea las páginas del manual de Apache y mire si usted puede implementar el uso de .htaccess.

ADMINISTRACION APACHE

En esta sección, miraremos los pasos requeridos para administrar un servidor web Apache. Veremos funciones de indexacion, hosts virtuales y archivos log.

Los siguientes tópicos son discutidos en esta sección:
- Indice del directorio
- Inclusión del Lado del servidor
- Hosts Virtuales
- Logging
- Formatos de log Personalizados

Indice del Directorio

Cuando una URL es presentada a un directorio, Apache busca un archivo especificado por la directiva de directorio index en srm.conf. Si no concuerda, Apache crea una pagina HTML dando una lista de los archivos en el directorio.

Por ejemplo:

Directorylndex index.html defau1t.html /cgi-bin/dontdothis

Esta entrada significa que Apache busca index.html, entonces default.html, luego ejecuta el script CGI dontdothis.

Puedes personalizar las entradas del directorio de páginas HTML, tales como los iconos, ultimos archivos modificados, las cabeceras y aquellos archivos que son utilizados. Utilice la siguiente directiva:

Fancylndexing On

Para ocultar archivos, utilice:
Indexlgnore Space separated-list-of-filename-patterns

Por ejemplo:
Indexlgnore */.??* *~ *# */HEADER* ‘/README* */RCS

Esta configuración es usualmente hecha en srm.conf.

La directiva IndexOptions tiene las siguientes opciones:

FancyIndexing	Compatible con versiones anteriores
IconsAreLinks	Iconos actuan como links
ScanHTMLTitles	Si los archivos están en HTML, abre y los lee como titulos (Tenga cuidado: esto incrementara la carga en el servidor)
SuppressDescription	No hay campo de descripción mostrado
SuppressLastModified	No hay muestra de modificaciones de fecha y tiempo
SuppressSize	No muestra el tamano de la información

Otras 2 directivas son también muy usadas:

HeaderNameyz99999	Aquellos archivos cuyo contenido es usado al inicio del listado de directorios
ReadmeName filename	Aquellos archivos cuyo contenido es usado al final del listado de directorios

Inclusión del Lado Servidor

La inclusión del lado servidor es un lenguaje preproceso para documentos HTML. Este agrega comandos especiales existentes en los codigos fuentes HTML, el formato es:
<!—#directive attribute ”val ue”- ->

Una directiva SM comúnmente utilizada es :
#Include

las cualidades para esta directiva son file o virtual:
<!—#i nclude vi rtual=”myheaderfile.html”—>
Este comando agrega el código desde myheaderfile.html a tu actual código HTML.

Otra directiva SSI frecuentemente utilizada pero peligrosa es:
#exec

Cualidades para esta directiva son cgi o cmd:
<]—#exec cmd=”lsi” >

Necesitas ser muy cuidadoso cuando permites a usuarios utilizar esta directiva en su servidor ya que ellos pueden remotamente ejecutar comandos en su servidor, y si no cambio el usuario desde root, los daños pueden ser severos.

Otras directivas SSI son:
#echo

Una cualidad var que es alguna variable de entorno CGI o una de las siguientes: DATE_GMT, DATE_LOCAL, DOCUMENT_NAME, DOCUMENTJJRI, or LASTMODIFIED

#fsize

El tamaño de los objetos referenciados por la cualidad[filej virtual!

#flastmod

La última fecha de modificación del objeto referenceado por los atributos del archivo virtual

#config

Varios atributos permiten la configuración de la salida de las otras directivas SSI

Hosts Virtuales

Frecuentemente, enfrentará un caso donde múltiples sitios web deben compartir un computador. Esto puede pasar en dos maneras diferentes. Una manera es correr 2 copias de Apache simultáneamente y tener cada una corriendo en servidores web diferentes. Esto es, para muchas partes, poco práctico y dificilmente utilizado. El método más común es ejecutar una copia de Apache y configurar el servidor para tener múltiples servidores y procesos de data independientes en el. Apache fué uno de los primeros programas en ofrecer host virtuales. Los hosts virtuales pueden ser configurados e implementados de diferentes maneras:

- **Hosts virtuales basados en nombre**
  Este es el mecanismo de host virtual más común y explota toda la capacidad del uso de dominios HTTP 1.1
- **Hosts Virtuales basados en IP.**
  Cada Web site esta disenado por diferentes direcciones IP. El servidor debe tener la capacidad de tener varias direcciones IP diferentes, una por cada host virtual.
- **Hosts virtuales mixtos.**
  Este concepto utiliza ambos hosts virtuales, basados en IP y basados en nombres.
- **Hosts virtuales basados en Puertos**
  Cada host virtual escucha puertos diferentes. La ventaja es que solamente una dirección IP o hostname es necesaria..

Los host virtuales basados en nombres son, en gran medida, utilizados en más frecuencia hoy. La única negativa del uso de este tipo de hosting es que aquellos clientes utilizando versiones viejas de navegadores podría no ser completamente compatibles.

¿Como Trabajan los Hosts Virtuales?

En el host virtual, varios hosts virtuales comparten la misma dirección IP. El servidor virtual posee una simple dirección IP asociada a el y múltiples nombres de dominio que apuntan a esa dirección IP, Por lo tanto, las peticiones al servidor son organizadas y el host virtual apropiado es contactado. El servidor virtual, por defecto, considera el directorio ~/www/htdocs del host virtual como el documento raíz, el directorio que contiene todos los archivos que pueden ser accesados a traves de un explorador web. Cada dominio que esta apuntando al servidor virtual puede ser asociado con un único y diferente directorio de documento raíz modificando el archivo httpd.conf en el directorio ~/www/htdocs . Por lo tanto, cuando una solicitud para un archivo llega al servidor virtual, un archivo diferente puede retornar dependiendo del nombre del dominio asociado con dicha petición.In virtual hosting, various virtual hosts share the same IP address.

El host virtual se puede proporcionar no solamente para los nombres completos del dominio, pero también para los nombres canónicos del dominio. Esto significa que el proveedor puede proveer el servicio no solamente a clientes con su propio nombre de dominio como www.client.com pero también nombres como www.client.provider.com .

Configurando los hosts virtuales

Para configurar el host virtual, es necesario que el nombre del dominio este registrado apropiadamente y el dominio sea agregado al servidor de nombres de la empresa de Internet o de lo contrario, el servidor virtual detendra su funcionamiento. Una vez el nombre del dominio esta registrado correctamente y agregado al servidor de nombres, el cliente puede configurar el nuevo host virtual sobre el servidor virtual.

Esto puede realizarse utilizando el comando vaddvhost. El comando vaddvhost modifica algunos archivos de configuración en el servidor y crea algunos directorios requeridos si estos no existen.

Los cuatro archivos de configuración principal en el servidor virtual son:

- **httpd.conf**

  Este es el archivo de configuración principal para el servicio web del servidor virtual. Contiene todas las directivas relacionadas a la operacion del servicio web. Las versiones actuales del servidor web Apache incorporan todos los detalles de la configuración en este archivo. Si estas utilizando una version más vieja, necesitarás utilizar los siguientes archivos también.
- **srm.conf**

  Este archivo contiene los directorios relacionados con los nombres y los recursos en el sistema de archivos.
- **access, conf**

  Este archivo contiene directivas que controlan las diferentes funciones que pueden realizarse por el explorador web.
- **mime, types**

  Esta archivo asigna los tipos de archivos basados en extensiones de nombres de archivos.

Ventajas del Host Virtual

El host virtual proporciona servicio web profesional rentable. Algunas veces el cliente puede tener su propia cuenta FTP que puede utilizar para mantener su sitio web.

El cliente puede tener su propio nombre de dominio como www.client.com o nombres canonicos del dominio como www.client.provider.coni. Finalmente el cliente puede recibir correo a su propio dominio como client@client.com, asi que no existe necesidad de conseguir una nueva dirección de correo, incluso si cambia de ISP.

Limitaciones del host virtual

Los hosts virtuales tienen limitaciones de espacio Web; Por lo tanto; es conveniente solamente para sitios web pequenos. Los hosts virtuales pueden manejar solamente de pequenas a medianas cargas. Si el número de peticiones se incrementa, la respuesta de la pagina web se retrasa. Si el servidor virtual recibe altos volumenes de tráfico, el número de hosts en el servidor virtual debe ser reducidos. Esto puede realizarse dando mayor tráfico al host virtual y moviendo algunos otros con un nivel de tráfico bajo.

El host virtual es posible por HTTP 1.1; por lo tanto, un navegador para ver los hosts virtuales debe tener soportar HTTP 1.1. Eso es, los host virtuales pueden ser visualizados por navegadores web estandares solamente. Los motores de búsqueda que no soportan el protocolo HTTP 1.1 no pueden indexar los hosts virtuales.

Ventajas de Servidores Dedicados Sobre los Hosts Virtuales

Un servidor dedicado puede ser también utilizado para grandes negocios. Todos los recursos de ancho de banda del servidor pueden ser utilizados por el propietario para varios propósitos.

El propietario del servidor dedicado dado tiene control completo sobre el hardware, el software de sistema operativo, y el software del servidor web .

El propietario tiene también control de administración completo del servidor. El sistema y las aplicaciones pueden ser configuradas para resolver los requerimientos unicos del sitio web.

Un servidor dedicado puede manejar miles de solicitudes diariamente y es la solución ideal para los sitios de gran tráfico.

Hosts Virtuales Basados en Direcciones

El host virtual basado en dirección envuelveAddress-based virtual hosting involves using multiple IP con un nombre por la dirección recibida en la misma máquina. La adición de un host virtual es hecha modificando el directorio de VirtualHost en el archivo de httpd.conf:

<VirtudlHosl 12.6.178.69> Server Name www.my- vi rtital host-name, coin DocimientRoot /home/vi rtual/public_html Transfer!og /home/vi rtual/Iogs/access_log Error Loci /homc/vi rtual /I ogs/error_l og </VirtiialHost>

Se recomienda que usted utiliza direcciones IP en lugar del hostname en el <VirtualHost> tag.

Virtual Hosts Basados en Nombres

Este esquema de host virtual permite que más de un host corra con la misma dirección IP. Esto se hace agregando nombres al DNS como un CNAME en la máquina. Un CNAME se puede pensar como un alias de DNS. la directiva NameVirtualHost en el archivo httpd.conf es asignado la dirección IP como es mostrada debajo:

NameVirtualHost 12.6.178.69

Después, fije la configuración para cada host virtual. Debes fijar la directiva ServerName para distinguir un host de otro.

<VirtualHost 12.6.178.69>
ServerName serverl.company.com
ServerAlias serverl
DocumentRoot /home/serverl/htdocs
ScriptAlias /home/serverI/egi-bin
TransferLog /home/serverl/logs/access .log </V1rtualHosl>
<VirtualHost 192.168,20-1.48> ServerName server2.company,com
ServerAlids server? Document Root /home/server2/htdocs ScriptAlias /home/server2/cgi-bin Transfer Log /home/server2/Iogs/accessJog </VirtualHost>

Almacenar en el log

El servidor Apache genera 2 archivos log por defecto: access .log y error.log. El formato para el archivo error.log especificado por las directivas CustomLog y LogFormat en los archivos httpd.conf. El formato actual del archivo log es fijado primero, y un nombre es dado en el siguiente formato:

LogFormst. "%h %1 %t \"%r \" %b " common

El nombre del archivo log es dado después, seguido por el nombre del ic del esquema del formato a seguir:

CustomLog logs/access_tog common

Si no es especificado como una ruta absoluta, se asume estar en la ruta relativa en ServerRoot.

Estas diferentes entradas en el archivo CLF son definidos como sigue:

host	Nombre del dominio del cliente (o número IP)
ident	Si IdentityCheck esta habilitada y el cliente ident esta ejecutándose, la identidad es mostrada
authuser	UserID Si es disponible como resultado de una petición para documentos protegidos
date	Fecha y hora de la petición
request	La petición recibidad desde el cliente
status	El código de estado volvio al cliente
bytes	El tamaño en bytes de los objetos devueltos al cliente

Formatos de Log Personalizados

Algunos ejemplos en formatos CLF incluyen:

LogFormat "%h %i %u %t \ "%r\" %>$ %b" common Log Format "3!{Referer} 1 -> W referer LogFormat "%{User-agentM" agent Custom Log /var/1 og/httpd/apache/access_l og common Custom /var/log/httpd/apache/referer_log referer CustomLog /var/log/httpd/apache/agent_log agent

Opciones de formatos de log personalizados

%...b:	Bytes enviados, excluyendo las cabeceras HTTP
%.,.£	Nombre del archivo
%...{VAR}e:	El contenido del la variable de entorno VAR
%.,.h:	Host remoto
%,..a:	Dirección IP Remota
	El contenido de VAR: línea(s) del cabecera en la petición enviada al servidor.
	Nombre de log remoto (desde el identd, si es proporcionado)
%...{VAR}n:	El contenido de VAR: Desde otro módulo
%...{VAR}o:	El contenido de VAR: línea(s) del cabecera en la respuesta
	El puerto canonic del servidor sirviendo la petición
	El ID del proceso del hijo que sirvió la petición
	Primera línea del estado de la petición;
	Time, en formato de log común, formato de tiempo
%...{FORMAT}t:	El tiempo, en ela forma dado por FORMAT, cual debe estar en el formato strftime(3)
	El tiempo que toma para servir la petición, en segundos
	Usuario remoto (desde auth; puede ser bogus si el estado de retorno (%s) es 401)
	La ruta al URL de la petición.
	El canónico ServerName del servidor sirviendo la petición
	El nombre del servidor respecto al ajuste de UseCanonicalName

Ejercicio 8-5: Administración de Apache

Este ejercicio proporcionará la oportunidad de llevar a cabo una sesion corta sobre administrar un servidor web Apache. Las soluciones a este ejercicio son proporcionadas en el Apéndice B.

1. Renombre su index.html y explore su servidor de nuevo. Usted consigue un listado de directorios ?

2. Accione la directiva Fancylndexing en su configuración de Apache. Reinicie Apache para que los cambios tengan efecto. Piensa usted que el indice generado por Apache luce mejor o peor con la directiva DancyIndexing habilitada ?

3. Utilice la directiva IndexOptions para cambiar los contenidos del indice:
   o IconsAreLinks
   :d SuppressSize
   3 SuppressDescription

4. Cree una pagina de myssi.html que utiliza un Server Side Include (SSI) para incluir un archivo sample.txt:

5. Como esto no esta trabajando, usted necesitara alterar la configuración; asi, renomre su arvhivo myssi.shtml e intente probar la pagina de nuevo.

6. Intente algunos otras directivas SSI, tales como:
   #echo #conf1g

7. Revise sus archivos de configuración y anote sus archivos de configuración y anote los nombres y ubicaciones de sus archivos log: los archivos AccessLog y KrrorLog.

8. Revise el contenido actual de estos archivos.

9. Edite el archivo httpd.conf y agregue registros adicionales descomentando las siguientes líneas:
CustomLog /var/1 og/tittpd/apache/referer_log refercr CustomLog /var/1og/httpd/apache/agent_log agent Stop and restart the server.
Entre al sitio con el navegador un número de veces con peticiones validas e invalidas y haga clic sobre un número de links. Revise los dos nuevos archivos log; que información que proporciona ?

OBTENIENDO EL CODIGO

Usted puede conseguir el código fuente para el servidor web Apache en http://www.apache.org o desde numerosas replicas a traves de Internet. Vaya a alguno de estos sitios y mire la sección de descargas. Usualmente, el código fuente de Apache esta almacenado en formato comprimido. Asi que después de descargar el código fuente, descomprimalo y extraiga los archivos de instalación utilizando el comando tar como es mostrado debajo:

tar -zxvf apache_1.3.19.tar.gz

Entonces cambie al directorio src en el directorio de instalación, puedes tener diferenctes codigos fuentes para el servidor Web Apache.

En esta sección, discutiremos la compilacion del fuente.

Compilando el Fuente

El siguiente paso es compilar el código fuente para completar la instalación. Primero tienes que configurar las estructura del proceso. Para hacer esto, edite el archivo de configuración .

Entonces defina su compilador C:
CC=gcc
Defina el tiempo adicional de compilacion ej:
EXTRA_CKLAGS-
Estas opciones dependeran de que configuración por defecto a ser.

El siguiente paso correra el script de configuración:

$ ./configure

Este crea los Makefiles necesarios basados en las opciones de configuración. Entonces ejecute el comando make:

Si todo fué configurado correctamente, entonces la instalación será satisfactoria y el servidor Web estara correctamente instalado.

Ejercicio 8-6: Compilar Apache

En este ejercicio, descargaremos los fuentes de Apache y crear el servidor Web. Si usted quiere permitir algunos de los módulos que su vendedor no incluyo. Necesitarás compilar de la misma manera para hacerlo en este ejercicio. La solución a este ejercicio son proporcionadas en el Apéndice B al final de este manual.

Este ejercicio instalara Apache en el árbol de directorio /usr/local/ , asi que no debería interferir con la instalación existente de Apache, el cual debe estar dentro de /usr/tree.

1. Descargue la última versión de los fuentes de Apache desde ftp.apache.org.

2. Desempaquete el código tarball.

3. Cambie al directorio fuente de Apache y configurelo. (Change to the Apache source directory and configure it. (Debe configurar para instalar /usr/local por defecto.

4. Construya elservidor Apache (Esto tomará algunos minutos crearlo).

5. Instale Apache.

6. Detenga el servicio viejo de Apache.

7. Inicie el nuevo servicio de Apache y asegúrese que trabaja. (Puedes necesitar fijar el campo ServerName en el archivo de configuración.

8. Asegúrese de que usted puede tener acceso al servidor del HTTP en localhost.

RESUMEN

En este capítulo, se introdujo el servidor web Apache. Los puntos claves que fueron discutidos: :

- Apache es el trabajo de un gran grupo de voluntarios en Internet.
- Apache tiene tres archivos de configuración principales:
  - httpd.conf es utilizado para configurar los atributos del servidor tales como el número de puerto y los usuarios que ejecutaran en el .
  - srm.conf instala la raíz del árbol de documento y funciones especiales como el servidor HTML analizo.
  - access.conf se utiliza para la configuración del directorio-nivel; permite o niega el acceso a ciertos directorios basados en los directorios en el archivo del conf.
- El funcionamiento de Apache implica el determinar de riesgos de la seguridad.
  - Asegúrese de que sus directorios de ServerRoot tengan los permisos apropiados.
  - Dé a Apache una identificación del usuario para funcionar con excepción de raíz.
  - Los scripts CGI pueden también ser un problema de seguridad si no configurados apropiadamente.
  - Permita la autentificación del usuario fijando directorios en el archivo de access.conf.
- Usted puede crear los hosts virtuales para actuar como más de un web server.
- Apache registra los logs de información de la conexión que usted puede mirar.
- Usted puede descargar y compilar la última version de Apache usted mismo.

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas están el el Apéndice A.

1. Nombre los tres archivos de configuración esenciales de Apache.

2. Que archivo necesitas modificar para permitir o negar acceso a un directorio en tu servidor ?

3. Por que Apache no soporta Secure Sockets Layer (SSL)?

4. Como usted inicia y detiene un servidor Apache?

5. Los servidores se pueden fijar hasta los anfitriones numerosos del servicio. Dé a ejemplo de los directorios del contorno general al sistema para arriba un anfitrión virtual.



E-MAIL: SMTP, POP, E IMAP

OBJETIVOS

Al finalizar este capítulo, usted estara preparado para efectuar las siguientes tareas:

- Básicamente operar y configurar el sendmail.
- Implementar sendmail.
- Implementar sistemas de correos POP3 e IMAP.
- Comparar y contrastar los siguientes sistemas de intercambio de correo: sendmail, smail, Qpopper, Mahogany, IMAP server y majordomo.
- Comparar los agentes de transferencia de correo sendmail, smail, y qmail.
- Describir los pasos necesarios para accesar email usando POP3 e IMAP en servidores de correo remotos.
- Describir el desarrollo e implementación de los programas mail y Pine en una red.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué es MIME?

2. ¿Cómo es que el SMTP (Simple Mail Transfer Protocol) maneja el correo?

3. ¿Cuáles dos partes conforman una dirección?

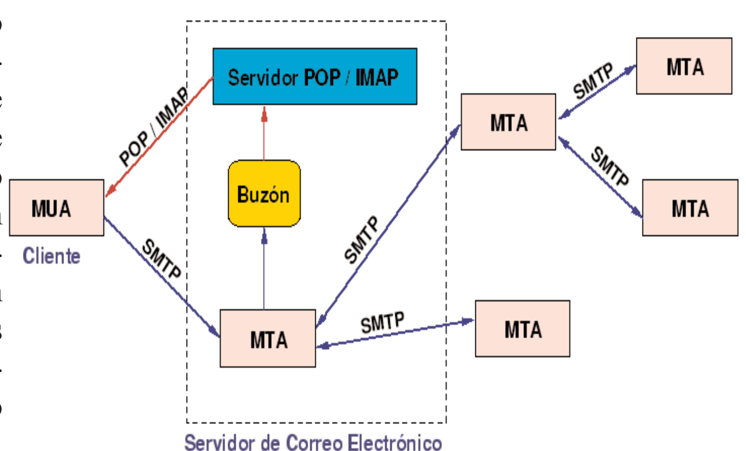
4. Describa un host de Correo.

5. ¿Cuál es responsabilidad de Agente de Usuario de Correo (MUA)?

INTRODUCCIÓN

En este capítulo estudiamos los conceptos principales de correo como es direccionamiento, enrutamiento y los demás componentes de una red de correo. Tomanos en consideración los protocolos de correo disponibles, como son SMTP, POP e IMAP. También mostraremos como una red de correo debe ser diseñada.

El correo electrónico clasifica como el servicio más utilizado de todos los que existen actualmente de arquitectura cliente-servidor. Gracias a este se tiene la posibilidad de comunicarse rápidamente con todo el mundo desde una estación de trabajo de forma muy simple y barata. Basta con tener un buzón en un servidor de correo correctamente configurado y una aplicación cliente para operar con dicho buzón. Un buzón de correo electrónico no es más que un archivo o un conjunto de ellos agrupados en un directorio donde se almacenan en cierto formato los mensajes que llegan.



Los MTAs (Mail Transport Agent) o agentes para la trasmisión de correo son aquellos programas servidores que permiten transportar el correo electrónico de una máquina a otra a través de la red. Como ejemplos de MTAs se pueden citar a Sendmail, Qmail, Postfix y otros. Todos hablan el mismo idioma, o sea utilizan un protocolo común para la comunicación: el SMTP (Simple Mail Transfer Protocol). El SMTP como su nombre lo indica es un protocolo muy simple orientado a caracteres y que permite el traslado de los correos tanto desde el cliente al servidor como entre servidores. Normalmente los servidores de correo utilizan el puerto 25 para comunicarse mediante el SMTP.

Por otro lado se encuentran los MUAs (Mail User Agents) que son los programas clientes que posibilitan a los usuarios manipular su mensajería. Estos programas son ejecutados directamente por los usuarios. Proveen facilidades para escribir los mensajes, enviarlos, descargar y leer los que llegan, organizarlos en directorios, hacer búsquedas, imprimirlos, mantener un libro de direcciones electrónicas, etc. Ejemplos de MUAs en Linux son los programas: pine, mail, mutt y elm. Para entornos gráficos están el Netscape Messenger y el kmail del entorno KDE.

Para enviar los mensajes, los MUAs se pueden conectar a un MTA determinado utilizando SMTP o escribir los mensajes directamente en el buzón si este es local. En cambio para descargar la mensajería un MUA puede hacerlo localmente o emplear alguna variante de dos protocolos fundamentales: POP (Post Office Protocol) o IMAP (Internet Message Access Protocol). Estos se diferencian en que el primero descarga la mensajería en la máquina local y el segundo permite manipularla directamente en el servidor y descargarla selectivamente. Un programa servidor o MTA no necesariamente acepta conexiones para descargar la mensajería que él manipula pues no siempre constituyen el destino final de la misma. Esta función constituye un servicio adicional, o sea si el sistema además de gestionar el envío de la mensajería mediante un MTA, permite descargarla debe ejecutarse también un programa servidor de POP o IMAP que brinde dicha funcionalidad. Actualmente en el caso de POP se emplea su versión 3 conocida como POP3.

El correo electrónico, E-Mail, es considerado por las grandes compañías el servicio de Internet más importante y de misión critica. La perdida de el servicio de e-mail lleva a estas compañía a serias perdida de comunicación y se traduce a perdidas millonarias si no se repara rápida y efectivamente.

FORMATO DE MENSAJES DE INTERNET

El RFC 822, "Standard for the Format of ARPA Internet Text Messages", describe el formato del mensaje y ha sido usado por todo lo amplio del Intenet. Dentro del dominio del Internet, es usado en sistemas tan diversos como los e-mails, noticias, y el www.

La razón principal del RFC 822 es definir un formato estandar del cabezal del mensaje. Este cabezal incluye algunos campos estándares suplidos por el que envía el mensaje como son:

To:	Recipiente del Mensaje en Internet
From:	Quien envía el Mensaje
Subject:	Tema del Mensaje
Date:	Fecha y Hora que el Mensaje fué Enviado

Este también define campos que deberán ser agregados por los saltos intermediarios, como es el campo Recibido:, para asistir en resolver problemas de correo y determinar rutas de correo.

Esto significa que debe ser razonablemente fácil escribir un sistema de correo que identifique, por ejemplo, el enviador y el que recibe el correo sin importar de donde el correo procede. De echo, una gran parte del RFC es dedicada a identificar tokens lexicales y posible contenido de cada campo para asistir a escribir tal programa.

MIME (Multipurpose Internet Mail Extensions)

Los mensajes definidos en el RFC 822 fueron perfectos para los dias de inicio del uso de los e-mails.Pero, este fué intensionado para mensajes de texto. Como tal, mensajes que contienen data multimedia, como son imágenes, sonido y video, no son mencionadas; no existe la capacidad para otro conjunto de caracteres que no sean ASCII, y el contenido de los mensajes está limitado a líneas cortas de ASCII de 7-bits.

Este problema ha sido enfrentado en los RFCs más recientes 1521 y 1522, "MIME Part One: Mechanism for Specifying and Describing the Format of Internet Message Bodies" and ""MIME, Part Two: Message Header Extensions for Non-ASCII Text. La meta de extender el componente del contenido de texto del RFC 822 de los mensajes de correo para que soporten múltiples objetos en un sólo mensaje, texto en conjunto de caracteres alternativos, mensajes de texto formateado con con multiple fuentes, y otros tipos de tipo de data multimedia. Esto se logra al especificar el mecanismo para codificar data en la líneas cortas ASCII de 7-bits que se mencionó anteriormente.

Los RFCs 1521 y 1522 convirtieron en obsoleto el contenido del MIME en los RFCs 1341 y el 1342. Todos los tipos de MIME deben ser aprobados por el IANA (Internet Assigned Numbers Authority). Una lista completa de todos los tipos de MIME esta disponible en el ultimo Assigned Number RFC.

ELEMENTOS DE TRANSFERENCIA DE CORREO

Para poder mejor entender las funciones ejecutadas por un sistema de correo, se a desarrollado una terminología común para describir los variados aspectos. La interfase que toma el contenido del mensaje y lo encapsula en un RFC 822, o MIME, es llamado un Mail User Agent (MUA), a menudo llamado simplemente un Agente de Usuario (UA) o simplemente un cliente de correo. Una vez se ha encapsulado un mensaje, el UA debe entonces prepararlo para que ser entregado al MTA (Mail Transfer Agent). Esto se lleva acabo o colocando el correo en una pila en el disco para que el MTA los recoja un poco más tarde, o pasarselo directamente al MTA (Agente de Transferencia de Correo).

Discutiremos los siguientes tópicos en esta sección:

• MTA (Agentes de Transferencia de Correo)
• SMTP (Protocolo de Transferencia de Correo Simple)

MTA: Mail Transfer Agent (Agente de Transferencia de Correo)

El MTA tomará los mensajes desde el User Agent o desde otro MTA y ejecutará varias tareas. Su tarea principal es la de interpretar la dirección del correo para decidir como es mejor enviarlo a su receptor. Si el que recibe no es un usuario local del sistema, el MTA deberá decidir si el puede llegarle hasta el receptor y si es asi por cual método. Una vez sea decidido el método de transferencia, este invocará lo que a veces es conocido como el Delivery Agent, como es el Simple Mail Transfer Protocol (SMTP), para transportar el mensaje más cerca de su destino final. Algunos MTAs, como lo es Sendmail, tienen uno o más Agentes de Entrega en ellos.

Si el MTA no puedo hacer contacto directo con el MTA del receptor, este pasará el mensaje a otro elemento de en la cadena de la transferencia de correo el Agente de Relevo (Relay Agent). Este Relay Agent o Host, de nuevo, analizará la dirección y continuará la cadena si el no puede entregar el mensaje directamente.

En el host que envía, el emisor, el usuario interactua con un User Agent para crear el mensaje que será enviado a otro usuario en el host que recibirá. Una vez el mensaje ha sido creado, este será pasado al MTA local, o directamente o vía una cola (queu) de correo, la cual el MTA recojerá cuando este listo para enviarlo.

El MTA entonces analizará el correo, interpretando cualquier alias, etc., y entonces decidirá donde enviarlo. Si el MTA no sabe como o donde llevar el correo, por ejemplo un usuario desconocido local o desconocido remoto, este retornará el correo al que envió con un mensaje de error.

Las opciones de aquí en lo adelante varían. Si el receptor es local para el host emisor, con esto quiero decir si el que envía y recibe el mensaje se encuentran en el mismo equipo, el MTA local colocará el mensaje directamente en el buzón (Mailbox) del receptor.

Si el receptor no está en el equipo local, el MTA, dependiendo como este configurado, intentará hacer contacto directo con el host remoto. Si descubre que el host remoto no está directamente accesible, puede ser configurado para que envíe el mensaje a otro MTA que si puede tener acceso. Este MTA es referido como un Relay Agent. Un Relay Agent (Agent de Relevo) es por lo general un MTA que conoce más el diseño de la red de correo y por esto puede remitir más´s eficazmente el mensaje que el MTA del emisor. Esto trabaja muy similar al enrutamiento del Internet Protocol (IP).

Un mensaje de correo puede pasar entre muchos Agentes de Relevo (Relay Agents) en su ruta a su destinatario. En cada MTA, si el mensaje no puede ser entregado inmediatamente, este por lo general es almacenado, esperando un segundo intento. Este comportamiento es conocido Store and Forward (Almacena y Reenvía). Típicamente, en el Internet, un MTA tratará de entregar un mensaje por 3 días antes de rendirse y devolver el mensaje.

Si en cualquier punto de la entrega el MTA no sabe como entregar el mensaje o se le agota el tiempo intentando entregarlo, será devuelto al emisor con un mensaje de error apropiado.

SMTP: Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo)

El protocolo principal de transferencia de correo usado que usa el TCP/IP es el SMTP. Este protocolo fué originalmente publicado en el RFC 821. El SMTP es un protocolo de punta a punta que usa el TCP como

transporte. El MTA que usa SMTP hara el intento de hacer una conexión al puerto 25 de host de destino (o al Host Receptor o al Agente de Relevo), donde espera encontrar otro MTA que también habla SMTP. Los dos MTAs entonces conversaran en ASCII para efectuar la transacción. Más de un mensaje puede ser enviado en una sola transacción.

El servidor de POP3 por defecto escucha por el puerto 110, mientras que IMAP lo hace por el 143 (ver /etc/services). Para comprobar el funcionamiento de uno u otro se pude utilizar el comando telnet. Ejemplo:

```
# telnet sion 110
Trying 192.168.100.2...
Connected to sion.
Escape character is '^]'.
+OK POP3 sion v2000.69rh server ready

# telnet sion 143
Trying 192.168.100.2...
Connected to sion.
Escape character is '^]'.
 OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=LOGIN] sion
IMAP4rev1
2000.283rh at Thu, 10 Jun 2001 16:11:40 -0400 (CDT)
```

DEPURAR SERVIDORES DE CORREO

Si un mensaje es devuelto sin entregar, existen unos pasos que se pueden tomar para determinar el porque. Si esta siendo enviado directamente desde su host a otro, entonces es buena idea probar la interconexión de la red usando el comando ping para asegurarnos que no es un problema de fallo de la red. Tome en cuenta que algunos firewalls no permiten el paso de paquetes ICMP (protocolo usado por ping).

Si la conexión de red se encuentra en buen estado, entonces puede probar usando telnet y conectarse al puerto 25, del host destino. Entonces puede proceder a escribir simples comandos SMTP para el MTA en ese host para ver si este rechaza el emisor o el receptor. Si usted aún no tiene suerte en encontrar el problema, puede revisar los archivos log para ver cualquier mensaje de error del MTA. Puede ser que usted tenga que usar una herramienta como nslookup para convertir el dominio de correo en un nombre de host; puede ver la sección más adelante en este capítulo donde nos referimos a este tema.

Los servidores POP pueden ser también diagnosticados usando el puerto número 110 y el IMAP usando el puerto 143.

Cuando un mensaje pasa a traves de un Relay Agent, el Agente le agrega una línea al cabezal del correo que dice quien es y cuando arribo. Si un mensaje llega pero toma mucho tiempo para llegar, usted podrá descifrar su ruta desde la información en estas líneas y detectar donde tardo la mayoría del tiempo.

ENRUTAR CORREO

La base de enrutar correo es la dirección. El MTA analiza la dirección para determinar la ruta que un mensaje debe tomar para acortar el viaje a su destino final.

La dirección está compuesta de dos partes: un nombre de usuario y un opcional nombre del equipo o nombre de dominio. Si sólo el nombre del usuario está presente, se asume que el correo es local. Ser local significa una computadora compartida o local el dominio, y por esto es entregado en un buzón local.

Si se específica el nombre de un equipo, el MTA intentará contactar el servidor de correo en el equipo remoto directamente. Si se usa un nombre de dominio, entonces el equipo que administra el correo de ese dominio debe ser determinado usando el DNS (Domain Name System).

Puede ser que también enfrente direccionamiento basado en ruta, aunque su uso es cosa del pasado, está conectado directamente a protocolos de punto a punto como es el uucp. Direccionamiento basado en ruta describe el conjunto de máquinas que un mensaje debe pasar para llegar a su destino:

maquina1!maquina2!maquina3!maquina4!maquina5

Ellas son muy flexible. Encontraras formas muy similar a esta en los headers de mensajes de noticias de Usenet; la dirección describe los servidores de noticias que el mensaje a pasado. Podemos usar una combinación de direccionamiento basado en rutas y basado en dominio un direccionamiento puede ser usado para mesclar las dos y crear una red híbrida.

Los siguientes tópicos son dirigidos en esta sección:
- Enrutamiento de Correo y alias del proyecto
- Enrutamiento de Correo y el DNS
- Rescribir la Dirección

Enrutamiento de Correo y los Alias

Un alias como el nombre los indica es un nombre alternativo. El equivalente a esto son las oficinas o puestos de correos y la redirección. Los alias son una parte muy útil del sistema de correo. Los alias son usados en tres instancias diferentes:

- Para proveer a un individuo con un nombre alternativo. Una convención es que las direcciones tengan la siguiente forma <inicial><apellido>. Esto es útil ya que también es costumbre en los nombres de usuarios. El usuario de nombre Carlos Brito, su usuario fuese cbrito y su correo es por convención cbrito@laempresa.net.
- Para reenviar el correo destinado a una máquina a otra para que el correo de un usuario le siga a una nueva posición.
- Para permitir que correo enviado a una dirección sea distribuido a muchos receptores; esto es llamado listas de correo o distribución.

Por lo general existen dos tipos de alias, personales y globales al sistema. Un alias personal será manejada por el User Agent del usuario. Con este tipo de alias, esta es resuelta antes de que el mensaje sea entregado al MTA. Estos alias solo pueden ser usados por el usuario que los definió. Los alias globales (o systemwide) si son resueltos por el MTA. Estos alias pueden ser usados por cualquier usuario local que utiliza ese MTA y los correos externos a la organización, de estos ejemplos convencionales son info@empresa.net y legal@empresa.org.

Esta es una manera de configurar un alias en un sistema GNU/Linux. Primero, debe editar el archivo /etc/aliases y agregue sus cambios. El archivo aliases relaciona una dirección de correo con una o más´s direcciones de correo. Después de estos cambios que el usuario efectué el administrador deberá ejecutar el comando newaliases. Esto crea una base de datos que el MTA lee cada vez que un mensaje de correo es enviado o recibido. Cuando un mensaje es recibido por el MTA y encontrado en la base de datos de los alias, entonces el encabezado es expandido y el correo es reenviado a su dirección de correo apropiada. Aquí le presentamos una entrada de ejemplo del archivo /etc/aliases:

```
valmis: valmis@abiertos.org, valmisdicarlo@hotmail.com, valmisdicarlo@gmail.com, arquitectura
arquitectura: bigwell@abiertos.org
cristian: c_nunez@abiertos.org
```

Enrutamiento de Correo y DNS

Al dia de hoy la gran mayoría de direcciones de correo de Internet son basadas en dominio o mejor dicho independiente del sitio. La dirección específica un buzón dentro del dominio, y no específico a ningún equipo. Los DNS asisten a enrutar el mensaje al equipo de destino con la ayuda del tipo de record MX. Este facilitario es útil cuando:

- El host final no puede ser alcanzado directamente debido a que este detrás del firewall. En este caso, el record MX direccionaría correo al mismo firewall, el cual entonces lo refeccionaría a traves y hacia la red local.
- El host final no puede ser alcanzado directamente debido a que se encuentra en una red diferente, detrás de una línea telefónica, o quizás una línea ISDN.
- El host final es una terminal sin discos y no ejecuta un MTA; el Record MX direccionaría el correo hacia un HUB de correo.
- El administrador desea que los usuarios envíen el correo directamente al nombre de dominio y a un host dentro de ese dominio. Esto permite que se pueda ocultar el host interno a una organización; un usuario externo no debe poseer este tipo de información.

El Record MX es estructurado de esta forma:

destination domain IN MX preference value forwarding domain

El valor de preferencia es usado cuando más de un Record MX apunta al mismo dominio. Los MTAs tratan de entregar a los dominios con los valores de preferencias más´s bajo.

Reescritura de Dirección

Hay ocasiones en las cuáles es necesario que el MTA tenga que reescribir la dirección escrita por el User Agent. Existen varias razones para que esto suceda, tales son cuando el MTA actúa como puerta de enlace entre dos diferentes protocolos de correo, para agregar el nombre correcto de dominio en una dirección local, o para esconder máquinas locales de la vista externa a nuestra red logrado con el reemplazo del nombre con un dominio genérico de la empresa.

PROTOCOLOS

Como hemos visto el Delivery Agent (Agente de Entrega) de correo preferido de TCP/IP es el SMTP. No siempre es factible para un host estar ejecutando SMTP. Puede ser que el sistema operativo no sea multitarea o que no tenga suficiente espacio en disco. También el SMTP presume que el host receptor va ha estar siempre en línea; de otra forma la entrega del mensaje fracasará (los MTAs permiten reintentos y timeouts). Claro esta esto no siempre es práctico para todos los hosts, en particular equipos portátiles y personales.

Dos protocolos que son particularmente útiles cuando el SMTP no es una opción viable son el POP3 y IMAP. Ambos protocolos sólo tratan con la recepción de los e-mails desde el servidor remoto y no proveen la opción del envío de correo, función que es llevada acabo por el SMTP o otro método.

Cubriremos los siguientes tópicos en esta sección:
- POP3
- IMAP

POP3

La versión 3 del Post Office Protocol permite que un computador cliente descargar los mensajes almacenados en un servidor POP3 a traves de TCP/IP o otro tipo de conexión de red. POP3 es muy popular y esta integrado en la mayoría de aplicaciones de correo electrónico y los navegadores de Internet más´s comunes. El correo es recibido y almacenado en el servidor hasta que el cliente o el receptor de correos del cliente revisa el buzón de entrada, muy similar a la manera en la que las oficinas de correo almacenan nuestros correos hasta que pasamos a retirarlo. El POP3 es capaz de descargar el nuevo correo depositado en el servidor, y el usuario tiene la opción de o removerlo después de la descarga o dejarlo para uso futuro. El usua-

rio también elije si se desconecta del servidor después de la descarga de sus mensajes a su equipo local. El usuario deberá reconectarse cada vez que desea enviar o revisar por nuevos mensajes.

POP3 es simple e ideal para los usuarios que no están conectados permanentemente a la red o en los casos que los recursos de red sean escasos, como es el ancho de banda y cuando las conexiones de Internet es costosa. Los usuarios pueden leer, componer y organizar sus correos fuera de la red. La simplicidad del POP3 crea cierta limitaciones, las cuáles discutiremos más adelante.

Si un usuario accesa su cuenta desde otro equipo puede ser que deje sus mensajes donde luego no pueda accesarlos u otra persona pueda tener acceso sin su permiso. Todos los correos son descargados al equipo local y el usuario puede perder control de su inventario. Las modificaciones hechas en un computador no son reflejadas en las otras que el ha accesado el mismo correo. Por estas razones siempre existe la posibilidad de duplicación de documentos en el uso de POP3.

IMAP

IMAP o Internet Message Access Protocol, que originalmente fué nombrado Interactive Mail Access Protocol. El cambio de nombre ocurrió para reflejar su predominante uso en el Internet. Es un protocolo estándar para el manejo de correo electrónico. En éste protocolo cliente / servidor el correo electrónico es recibido y almacenado para tí en el servidor del proveedor. Permite consultar únicamente los encabezados (remitente, titulo, tamaño) del mensaje antes de decidir si descargar, almacenar o eliminarlo. El protocolo IMAP permite también la creación y manipulación de folders o buzones en el servidor. IMAP es una alternativa sofisticada desarrollado para compensar algunas de las limitantes del protocolo POP3.

A diferencias del POP3 los usuarios pueden accesar sus correos desde cualquier punto, sin tener que descargar los archivos. De esta manera no existe ningún tipo de repetición involuntaria de archivos ya que independientemente de cuantas máquinas diferentes se accesa el correo cada instancia manipula el mismo buzón.

El IMAP si requiere conexión continua al servidor mientras se manipulan los mensajes de correo. Pero, también soporta la lectura de correos fuera de línea y los usuarios deciden si descargan solo los cabezales o el contenido completo para su lectura fuera de línea.

Cuando un usuario primero accesa el servidor IMAP, los archivos no son descargados inmediatamente. Sino que interactivamente el modelo cliente /servidor entra en efecto y el cliente le pide los cabezales o los contenidos al servidor de ciertos mensajes o realiza búsqueda bajo cierto criterios. Luego pasa a que los mensajes son marcados con varios distintivos (flags) de estado (Ej., eliminados, leídos, sin leer). Los mensajes permanecen en el servidor hasta que el usuario los elimine. Asi es que IMAP está diseñado para permitir la manipulación de los mensajes en buzones remotos como si fuesen locales. Dependiendo del cliente de IMAP, el usuario puede guardar los mensajes directamente en la máquina cliente o en el servidor de correo.

Para establecer un servidor IMAP (los distros oficiales lo incluyen pero si desea una versión diferente tendrá que descargarlo e instalarlo):

El software lo puedes obtner en: .
ftp://ftp.cac.washington.edu/imap
Requerimientos
- **Para utilizar IMAP debemos tener instalado un MTA(Mail Transport Agent) como sendmail.**
- **Espacio en disco durante la instalación 27 MB y para instalación 5 MB.**
- **Configurar un cliente de correo.**

Instalación

1.- Descomprimir y destarear el archivo.
$ unzip imap.tar.Z
$ tar xfv imap.tar

2.- En el directorio creado, teclear:
$ make six
six es el código para Linux. Para otros sistemas ver el Makefile.

3.- Convertirse en superusuario
$su -

4.- Copiar los archivos.
cp ipopd/ipop2d /usr/local/sbin
cp ipopd/ipop3d /usr/local/sbin
cp imapd/imapd /usr/local/sbin

5.- Editar el archivo inetd.conf para descomentar las líneas siguientes:
#vi /etc/inetd.conf

pop
stream tcp
nowait root
/usr/sbin/tcpd
/usr/local/sbin/ipop2d

pop3
stream tcp
nowait root
/usr/sbin/tcpd
/usr/local/sbin/ipop3d

imap
stream tcp
nowait root
/usr/sbin/tcpd
/usr/local/sbin/imapd

6.- Editar el archivo /etc/services para descomentar las líneas siguientes:
vi /etc/services

pop
109/tcp
pop-2
postoffice # POP version 2

pop
109/tcp
pop-2

pop3
110/tcp
pop-3
POP version 3

pop3
110/tcp
pop-3

imap
143/tcp
imap
IMAP
imap
143/tcp
imap

7.- Copiar los archivos.
cp c-client/c-client.a /usr/local/lib/libc-client.aÊ # cp c-client/rfc822.h /usr/local/include
cp c-client/mail.h /usr/local/include # cp c-client/linkage.h /usr/local/include

8.- Iniciar el demonio de inetd.
#kill -HUP PID-inetd

SEGURIDAD DE E-MAIL

Los mensajes de correo son transmitidos vía el Internet como texto simple. Además, conversaciones de SMTP son no autenticados, y las comunicaciones POP son autenticadas usando sólo una contraseña, las cuáles también son en texto simple. Con redes tan grandes como el Internet y data viajan a traves de muchos enrutadores, es posible interceptar mensajes en sus rutas. Los mensajes también se pueden simplemente perder. Esto puede ser causa de un direccionamiento incorrecto o estar siendo direccionado incorrectamente por uno de los hosts intermedios.

En la actualidad no existe un estandar oficial de encriptación de data en el Internet, pero no se duda que el futuro traerá uno que otro.

La lleva PGP (Pretty Good Privacy) es la más usada para asegurar sistemas de correo. Este utiliza el algoritmo RSA de llave publica para distribuir llaves de encriptación y autentificar mensajes. Una implementación de dominio público existe cuando; con este un usuario debe primero encriptar su mensaje y entonces enviar su salida. Algunos clientes soportan las llaves PGP directamente. Los usuarios deben generar sus conjuntos de llaves que contienen las llaves públicas de los usuario con lo que ellos desean comunicarse.

Se ha propuesto una variedad de protocolos de seguridad para el correo electrónico en Internet, pero sólo uno o dos han recibido cierto uso extendido. El correo enriquecido con carácter privado (PEM-Privacy-enhanced mail) es un estándar de Internet para asegurar al correo electrónico usando llaves públicas o simétricas. El uso de PEM ha descendido, aunque existe un RFC que lo define y es un esfuerzo en conjunto de IETF, pero versiones iniciales usaron algoritmos de los Estados Unidos DES. Tampoco no está diseñado para manejar el moderno correo electrónico de multipartes soportado por MIME, además de que requiere una jerarquía rígida de autoridades de certificación para emitir llaves. Secure MIME (S/MIME) es un estándar más nuevo que se ha propuesto basado en el standar existen MIME pero introduce extra headers para especificar mecanismos de encriptación y autenticación. Usa muchos de los algoritmos criptográficos patentados y cuyas licencias corresponden a RSA Data Security Inc. S/MIME depende de certificados digitales, por ello también depende de algún tipo de autoridad de certificación, ya sea corporativa o global, para asegurar la autenticación.

INTEGRACIÓN DE SISTEMAS DE CORREO PROPIETARIO

Muchas compañía utilizan sistemas de correo propietarios. Exchange, Lotus Notes, entre otros son los más populares pero existen otros. Estos sistemas ofrecen un número de ventajas sobre el correo de Internet, como son mensajes seguros y autenticados, prueba de recepción y horario. Extrañamente es todo estas extras funciones que se han convertido en su propio enemigo y ha permitido que el correo de Internet

los venza.

Los sistemas de correo de Internet son altamente escalable; el Internet es en la actualidad más grande que cualquier empresa y realmente para su tamaño funciona muy bien. El crecimiento del Internet ha significado que muchas compañías desean intercambiar mensajes de correo entre sus aplicaciones privadas y los correos de Internet. Afortunadamente todos los sistemas de correos privados brindan cierto soporte para llevar esto acabo, a traves del soporte de los protocolos de transferencia que incluyen el SMTP, DNS, POP y muchos incluyen hasta el IMAP. Conectarse a los correos de Internet es simplemente un asunto de correctamente configurar los componentes necesarios. Los administradores deben siempre tomar en cuenta que los usuarios soamente podrán intercambiar mensajes de RFC 822. La conversión de formatos propietarios a los básicos de Internet puede resultar en perdida de información; MIME proveéis una solución pero no es universal aún.

La adopción de redes internas o Intranets basadas en TCP/IP y los protocolos estándares, en particular el SMTP; reduce muchos de los problemas asociados con la integración de sistemas propietarios de correo. Además los protocolos de Internet también empiezan a ofertar las mismas funcionalidad de los sistemas propietarios. Una solución basada puramente en el Internet puede ser muy atractiva para cualquier compañía.

DISEÑO DE UNA RED DE CORREO

Sin la aplicación de un diseño o plan, su red de correo puede rápidamente deteriorar hasta el punto de convertirse inservible. Para el diseño de una red de correo estos son algunos puntos sobre los cuáles se puede construir algo más duradero:

Mail host El host de correo es el punto central del enrutamiento del correo o la oficina donde se organiza (ordena) el correo. Equipos individuales que no saben donde enviar sus correos los enviaran aqui, la cual tiene toda la información necesaria de toda la red de máquinas y dominios contactables por la red de correo.

Mail Relay Un Relevo de Correo, Reenvía o releva, el correo cuyo destino final es fuera de la organización. También puede ser quien recibe los correos externos. También puede recibir correos entrantes. El relevo conversa a otro equipo igual que el (peer) que habla su mismo lenguaje (por ejemplo el SMTP)

Mail Gateé Un Gateé (pasarela) También puede reenviar correos a destinos fuera de la organización. Pero, el Gateé forma un vinculo entre los dos sistemas de correo. Por ejemplo, si un correo va a ser enviado a una VAX ejecu tandose cobre un VMS, el Gateé maneja los protocolos de conversión para enviar el mensaje al correo VMS. El VAX recibirar los mensajes como correos VMS y no ejecutara ninguna conversión.

Mail Server Un mail server almacena buzones de correos para sus clientes. Los mensajes de correos entrantes se mantienen en el servidor hasta que el cliente lo retire. Cualquier máquina puede ser un servidor de correo pero si toma en consideración el requerimiento de espacio en disco duro.

Mail Client Los clientes retiran sus correos desde el servidor. Este También es conocido como Mail User Agent.

Note que estos son componentes funcionales y no necesariamente equipos individuales. Por ejemplo, una máquina única puede ser configurada para ejecutar ambos el host de correo y la función de relevo. También considere una locación remota con solo una máquina; debería ejecutar las funciones de correo, host, relevo, servidor, y cliente.

La ocurrencia incrementada de spam, o correo no solicitado en el Internet, convierte en esencial que el servidor ofrezca la funcionalidad de AntiSpamm. Este no debe aceptar correos no deseados desde un sistema remoto para los host fuera de su dominio y debe almacenar la dirección IP del servidor remoto.

CORREO/MAIL

El utilitario mail es usado para enviar y recibir mensajes de correo. Este fué originalmente desarrolado para ÓNIX BSD y fué simplemente bautizado con el nombre de mail. La mayoría de sistemas GNU/Linux incluyen una version modificada de este utilitario, de nombre mailx, para enviar y recibir mensajes. En sistemas en los cuáles mailx esta instalado, mailx por lo general es un alias a mail. El utilitario mail es configurado a traves de un archivo de nombre .mailrc, el cual proveéis características como la de crear

alias.

En esta selección discutiremos los siguientes tópicos:
- Enviar un Mensaje
- Editar un Mensaje
- Recibir un Mensaje
- Desplegar un Mensaje
- Responder a un Mensaje
- Enviar un Mensaje Nuevo
- Salir del Shell de Mail

Enviar un Mensaje

Para enviar un mensaje usando el utilitario mail, escriba mail en el prompt seguido por la dirección del receptor. Al presionar ENTER, se le pedirá que digite el tema del correo. Luego de someter el tema del correo, pulse ENTER de nuevo y esto le colocará en el modo de entrada, y de aqui en lo adelante lo que digite será tomado como el cuerpo del mensaje. Una vez completado el cuerpo del mensaje, presione CONTROL+D. Se le presentará la opción de entrar una Copia Carbon (Cc:) del mensaje para enviarla. Si presiona ENTER sin escribir una dirección de correo adicional en el prompt de Cc:, al presionar ENTER se enviará el mensaje. Tome este ejemplo:

```
$ mail  ivelis
Subject  : Saludo!
Como estas?
 ^D
Cc  :  miguel
$
```

En este ejemplo, el receptor esta en el mismo equipo que el emisor. Si el receptor está en otro host, entonces escribirías la dirección completa de receptor asi: ivelis@abiertos.org. En este ejemplo el cuerpo del mensaje se escribió en el shell del mail. Pero, también podemos usar un archivo ya escrito para enviarlo al receptor con el uso de redirección. Entonces nuestro mensaje empezaría asi:

```
$ mail   ivelis  <  archivo.txt
```

En este caso, el contenido de archivo.txt es enviado usando el operador de redirección. El archivo.txt puede ser creado usando cualquier editor de texto estandar.

Para enviar correo a más de un usuario a la vez, escriba la dirección de uno tras otro en una sola línea:

$ mail ivelis miguel cesar ... etc ...

En este ejemplo, el mismo correo será enviado a Ivelis, Miguel, Cesar, ...etc...

Para guardar una copia del mesaje enviado, incluya el nombre con el cual lo desea guardar. Si lo desea guardar en el mismo directorio en el cual se encuentra entonces debe precedirlo con un punto y una barra seguido por el nombre que desea asi:

$ mail ivelis ./nombre.archivo

Si desea almacenarlo en otro directorio que no es el de trabajo actual entonces debería anteponerle la ruta completa del directorio donde desea guardarlo.

Editar un Mensaje

Comúnmente un mensaje tiene dos partes el cabezal y su cuerpo. El encabezado consiste de la dirección de destino y el tema, mientras que el cuerpo de texto del mensaje consiste de la información que el usuario desea transmitir a la dirección de destino.

Un conjunto de comandos conocidos como los comandos tilde que usados para editar tanto el encabeza-

do como el cuerpo del mensaje. Un comando tilde es una combinación de una tilde (~) seguido de un comando de un carácter. Esta combinación no es tomada como parte del mensaje, para poder incluir una tilde en un mensaje deberá escribir dos tilde. Los comandos tilde empleados en el encabezado son usados para cambiar la información en el encabezado y la dirección, mientras que los usados en el cuerpo redesplegan, guardan e invocan un editor para modificar el texto del cuerpo.

Empiece con un comando de tilde en el cuerpo del mensaje. Sabemos que no existe características que faciliten el movimiento del cursor a una posición deseada, insertar o borrar frases. El único tipo de corrección permitida es borrar los caracteres que están inmediatamente a la izquierda del cursor. Para poder prover capacidad de edición usamos el comando ~v. La ejecución de este comando invocará el editor vi. En el modo de edición de vi, el mesaje escrito aparecerá como el texto a ser editado. Luego de editar el texto el usuario podrá salir del editor vi, escribiendo el comando zz. Luego de salir del editor vi note que el mensaje editado no es mostrado. La palabra continuar aparecerá en la pantalla. El usuario puede continuar escribiendo más texto o ejecutar otro comando de tilde.

A pesar de que el comando ~v es útil para editar mensajes, no permite ver el mensaje editado. Para poder ver el mensaje editado use el comando ~p. El comando ~p mostrará el mensaje previamente escrito. Por ejemplo, en el siguiente comando mail, al ejecutar el comando ~p nos muestra el mensaje previamente escrito:

```
$ mail ivelis
Subject   :  Hola!
Hola, Ivelis, como estas?
     ~p
    .....................................
     Message contains  :
To:  Ivelis
Subject : Hola!
Hola, Ivelis, como estas?
(continua ....................)
^D
EOT
$
```

Hay comandos tilde que son usados para guardar mensajes a un archivo y para insertar el contenido de un archivo en un mensaje. El comando para saltar el mensaje a un archivo es ~w. El uso de este comando es ilustrado en el siguiente ejemplo:

```
$ mail ivelis
Subject   :  Hola!
Hola, Ivelis, como estas?
~w  archivo.guardar
“archivo.guardar”  2/14
```

Al ejecutar el comando ~w archivo.guardar, el contenido del mensaje es guardado en el archivo de nombre archivo.guardar.

El comando para insertar el contenido de un archivo en un mensaje es ~r. El uso de este comando es ilustrado en el siguiente ejemplo:

```
$ mail ivelis
Subject   :  Hola!
Hola, Ivelis, como estas?
~r  archivo.txt
“archivo.txt”  4/17
~p
......................................
```

Message contains :
To: ivelis
Subject : Hola!
Hola, Ivelis, como estas?
Hola, Ivelis, como estas?
(continua.....)
^D
EOT
$

Al ejecutar el comando ~r archivo.txt, el contenido en el archivo archivo.txt es anexado al cuerpo del mensaje que estamos confeccionando.

hay comandos tilde para salir del utilitario mail, si es que el usuario por alguna razón desea abortar el envío del mensaje. Ambos comandos ~x y ~q son usado para abandonar el utilitario mail. En el caso del comando ~q, el mensaje se almacenará en un archivo de nombre dead.letter.

La edición de texto puede ser llevada acabo con el uso de filtros. El comando tilde usado para efectuar esto es ~|. (El carácter | es conocido como la tubería o pipe). Este comando toma un filtro como argumento, asi que más de un filtro puede ser usado en un mensaje. El cuerpo del mensaje se convierte en la entrada estandar. El filtro usa esta entrada para producir un mensaje de salida que reemplaza el mensaje original. En el ejemplo que sigue usamos el filtro sort:

$ mail ivelis
Subject : Listado usuarios
ivelis
cesar
miguel
cristhian
elvyn
raffy
~| sort
~p
El filtro sort nos da de salida el siguiente texto:
Message contains:
To: ivelis
Subject: Listado usuarios
cesar
cristhian
elvyn
ivelis
miguel
raffy

Otro filtro muy útil es fmt. Este filtro es usado para dar formato a un texto sin formato y convertirlo en líneas de texto estandarizado. En el siguiente ejemplo ilustramos el uso de este filtro:

```
$ mail ivelis
Subject : Hola!
Hola Ivelis, como estas?
              Hace tiempo que no se de ti!
  Si tienes tiempo escribeme.
~| fmt
(continua.........)
~p
```

El filtro fmt rinde la siguiente salida de texto, el cual reemplaza el texto en el mensaje original:

$ mail ivelis

Subject : Hola!
Hola Ivelis, como estas?
Hace tiempo que no se de ti!
Si tienes tiempo escribeme.
(continua.........)
^D
EOT
$

Los comandos ~ son usados para editar el cabezal del mensaje También. El cabezal del mensaje tiene cuatro partes principales: el tema (subject), lista de direcciones, Cc, y Bcc. De estas Cc y Bcc son opcionales.

Si el usuario desea agregar más direcciones mientras el escribe el mensaje, entonces el comando tilde ha ser usado es ~t. En el siguiente ejemplo ilustramos el uso de este comando:

$ mail ivelis
Subject : Hola!
Hola Ivelis, como estas?
~t jose
~p

Con el comando ~t hemos agregado a Jose como un receptor adicional de nuestro mensaje:

Message contains:
To: ivelis jose
Subject : Hola!
Hola Ivelis, como estas?
(continua....)
^D
EOT
$

Un usuario puede cambiar el tema con el comando ~s:

$ mail ivelis
Subject : Hola!
Hola Ivelis, como estas?
~s saludos ivelis
Message contains:
To: ivelis
Subject : saludos ivelis
Hola Ivelis, como estas?
(continua....)
^D
EOT
$

Para enviar un mensaje a alguien que su dirección no debe ser impresa al final del mensaje, digite la dirección del usuario receptor después del comando ~b, el cual crea una copia de carbon oculta. Las direcciones que se colocan después de comando ~c apareceran al final del mensaje para todos los receptores del mensaje. Una copia es creada con el comando ~c.

$ mail ivelis
Subject : Hola!
Hola Ivelis, como estas?
~c cesar miguel
~b jose
~p
Message contains:

To: ivelis
Subject : Hola!
Cc: cesar miguel
Bcc: jose
Hola Ivelis, como estas?
(continua....)
^D
EOT
$

Recibir un Mensaje

Otro aspecto muy importante del utilitario mail es el de recibir los mensajes entrante. Podemos entrar en el shell de mail y verla solo con escribir el comando mail en el prompt. Si no hemos recibido algún correo para el usuario con el que nos encontramos ingresado (ivelis) nos desplegará el siguiente mensaje:

No mail for ivelis

Si el buzón no está vacío, entonces al entrar el shell de mail se desplegará el resumen de cada uno de los mensajes. El resumen del encabezado contiene varios campos. El primer campo es el estatus. Existen dos opciones para este campo; N, que significa un nuevo mensaje, y U que significa mensaje no leído. El segundo campo es el número del mensaje, y el tercero es la dirección de quien envía. El cuarto campo representa la fecha y hora, y el quinto representa el número de líneas y caracteres en el mensaje.

```
$ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/ivelis": 1 message 1 new
  U  1 cesar@abiertos.lo  Sat Mar 19 09:47  15/492   "hola"
>N  2 jose@abiertos.lo  Sun Mar 20 10:47  15/392   "adiós"
&
```

El símbolo (>) marca el mensaje actual. Si no se incluye el número del mensaje, entonces el mensaje con el marcador será tomado como el mensaje actual. Los mensajes pueden ser visto con la ayuda de la lista de los mensajes. En este ejemplo anterior podemos ver los mensajes con simplemente digitar 1 o 2 en el prompt. Existen caracteres especiales como es el ^ y el $. El carácter especial ^ representa el primer mensaje y el carácter $ es el último. Asi que la lista de los mensajes puede ser especificada con ^-2 o 1-$. Los mensajes basados en en direcciones de los emisores o el tema del mensaje pueden ser visto con simplemente escribir la dirección en el prompt de &. Esto desplegaría todos los mensajes enviados por el usuario con esa dirección. Escribir el carácter (/) seguido por un tema, desplegaría la lista de mensajes que igualan este tema. Escribir el carácter (:) seguido por "n" nos devuelve de los mensajes nuevos, y los dos puntos seguido por (:u) nos devuelve la lista de todos los mensajes sin leer. Escribir (p : n) nos desplegará todos los mensajes nuevos.

Desplegar un Mensaje

Podemos desplegar los mensajes con el comando mail. El usuario puede hacer los mensajes en orden pulsando ENTER. Para ver un mensaje en particular, escriba el número del mensaje. El comando mail también proveéis una forma de referenciar y desplegar los mensajes de acuerdo a su posición, con respecto al mensaje actual. Para desplegar el mensaje anterior al actual, se usa el comando (-). Por ejemplo si el usuario se encuentra en el comando número 7, entonces si escribe (-5), se desplegaría el mensaje 2. Para desplegar el mensaje después del actual, entonces podemos usar el comando (+). Por ejemplo, si estamos actualmente en el mensaje número 3, el comando (+4) desplegaría el mensaje número 7. Para poder desplegar un rango de mensajes, los comandos p y t son usados. Por ejemplo, escribiendo (p 1 3), se despliegan los mensajes 1 y el 3 solamente. Escribiendo el comando (p 1-3), todos los mensajes del 1 al 3 son desplegados.

Eliminar y Recuperar un Mensaje

Para poder eliminar un mensaje, el comando (d) es utilizado. El comando d toma un número como argumento y elimina el mensaje con ese número. Por ejemplo, el comando d 2 elimina el segundo mensaje. Para poder eliminar un un rango de mensajes, deberá especificar el rango con el comando d. Por ejemplo, el comando d 3-7 elimina los mensajes 3,4,5,6, y el 7. Si usamos el comando d sin ningún argumento elimina el mensaje actual.

Si un usuario desea deshacer la eliminación, entonces usamos el comando (u). El comando u sólo puede recuperar un mensaje siempre y cuando el usuario no haya salido del shell. El comando u es similar en sintaxis al comando d. Por ejemplo, el comando u 3, recupera el tercer mensaje, mientras que el comando u 3-5 recupera los mensajes 3,4, y el 5.

Responder un Mensaje

Para responder un mensaje, el utilitario mail tiene dos comandos, (r) y (R). Ambos toman como argumento un número, y presentan un prompt al usuario para escribir el texto a responder. El titulo del cabezal empieza con Re: para indicar respuesta. La única diferencia entre los comandos r y R es cuando el usuario recibe un correo que ha sido enviado a multiple usuarios, el comando r le envía respuesta a todos los usuarios, mientras que el comando R solo le responde al emisor.

Enviar un Mensaje Nuevo

En ves del comando mail, podemos usar el comando m para enviar un nuevo mensaje.El comando &m ivelis funciona similar a mail ivelis. También el comando m tiene todas las versatilidad del comando mail, como son el comando tilde, etc.

Salir del Shell de Correo

Para poder salir de shell de mail, debes usar el comando q. En el momento que invocas este comando, todos los mensajes leídos se almacena en un archivo, llamada mbox, la cual se localiza en el directorio home. Todos los mensajes que no leyó son retenidos en el buzón o mailbox. Si el usuario desea tener los mensajes leídos en el buzón, tendrá que usar el comando pre. El shell de mail puede ser abandonada con el comando x. El comando x difiere del comando q en que los mensajes eliminados durante la sesión serán eliminados.

EL SENDMAIL

El Agente de Transporte de Correo (MTA) más usado en GNU/Linux es Sendmail. Un MTA es un cliente que efectúa el proceso de enviar correo; pero los MTAs no son responsable por buscar, componer o organizar emails. Un cliente diferente conocidos como MDA (Mail Delivery Agent) es el responsable de recibir los correos, mientras que el MUA (Mail User Agent) es el responsable de componer y organizar el correo. Ejemplos de MUAs en GNU/Linux son bajo ambiente X11 ximian evolution, Qpopper, kmail y bajo ambiente de consola podemos mencionar mutt y pine entre otros. El Sendmail es compatible con los MUAs y MDAs más populares. Esto lo que significa que los correos escritos en Pine pueden ser reconocidos y enviados por Sendmail. También, los emails enviados por sendmail pueden ser buscado por fetchmail, el MDA más popular.

Escrito por Eric Allman y modificado por SUN y la Hewlett-Packard, sendmail ha evolucionado desde su creación en el año 1980. Sendmail viene preinstalado con configuración por defecto en la las mayorías de distros de GNU/Linux. Por lo general puedes encontrar a sendmail en el directorio /usr/lib. Hoy en día la última versión de sendmail es 8.xy.z, la cual puede ser descargada desde el sitio Web oficial http://www.sendmail.org

En esta siguiente sección discutiremos los siguientes tópicos:
• Configurar Sendmail
• Ejecutar Sendmail
• Diagnosticar Sendmail
• Smail
• Qmail
• Qpopper
• Mahogany

EL SENDMAIL

En esta sección analizaremos los archivos utilizados por sendmail asi como las listas de correos y el comando newaliases.

Sendmail.cf

Cuando ejecutamos sendmail, el primer archivo que este lee es el archivo /etc/sendmail.cd. Este archivo se encarga de decirle a sendmail donde se encuentran todos los archivos que este necesita para funcionar adecuadamente. Una de las funciones principales de sendmail.cf es ser el almacén de información de sendmail. Puedes ver a sendmail como un gran árbol. Una de las ramas de sendmail.cf son los alias, sendmail.st, sendmail.hf y el directorio de cola (qeue). El archivo sendmail.cf contiene las reglas usadas para escribir las direcciones de correo. Algo muy importante de entender antes de modificar este archivo es que esta basado en línea. Mucho cuidado de no crear líneas vacías o con tabulados. Esto creará problemas que a veces son imposible de diagnosticar.

Para implementar estas secciones, cada una de las líneas del archivo de configuración comienzan por una de estas letras, según la función que realizan:

#	Comentario
espacio	Continúa desde la línea anterior
tab	Continúa desde la línea anterior
S	Definición de un conjunto de reglas de reescritura
R	Definición de una regla de reescritura
D	Definición de una macro
C	Definición de una clase
F	Extraer los tokens de una clase de un archivo
O	Definición de opciones
H	Formato del texto de la cabecera
T	Usuarios de confianza (Trusted Users)
P	Precedencia y prioridad para los mensajes
M	Definición de un agente de transporte
V	Versión del archivo de configuración

Es aconsejable que cada vez que va a modificar su archivo de configuración de sendmail.cf efectúe una copia de seguridad de este archivo.

Sendmail.cw

El archivo sendmail.cw almacena todos los alias y desplega todos las entradas de dominios virtuales. Los archivos domain.com, server1.domain.com y mail.domain.com se encuentran en el directorio /etc, el cual le los nombres locales de host en su sistema. El archivo sendmail.cw requiere que usted coloque todos sus alias de su equipo.

Este es un ejemplo de agregar una entrada en el archivo sendmail.cw:
FEATURE (‘use_cw_file’)

El siguiente comando convertir UUCP a una base de datos:
FEATURE (‘use_cw_file’)

Alias

Los sistemas de correo permiten dos tipos de alias: alias de correo del sistema y del sistema de correo de usuarios. Estos alias de correo no deben confundirse con los alias de comandos discutidos anteriormente cuando aprendió manejo de comandos.

Los alias del sistema se encuentran en el directorio /etc. Los alias del sistema siempre crean un archivo sendmail, el cual se encuentra en el directorio /usr/lib. La gran mayoría de administradores utilizan el comando newaliases para actualizar la base de datos de sendmail. Algunas de las alias más comunes son root, postmaster, ftp, Web, news y Mail. Los alias no siempre apuntan a una dirección única pero a multiple. Los alias del sistema deben siempre enviar a una dirección válida de correo.

Ejemplos:
postmaste: root
Mail-daemon: root
news: root
systems: ivelis, cesar
assitente: root, miguel, carlos, jose

El asistente es un alias de root, miguel, carlos y jose, quienes recibirán un correo dirigido a asistente además de el ellos.

Otros alias conocidos son para el sistema de listservs. En listservs, usted se suscribe a una lista simplemente escribiendo “subscribe” en el cuerpo del mensaje y envía el correo al listserv. El comando para encontrar todos los listservs en el sistema es **majordomo@listserv.xx.xxx.**

Los alias de correo de los usuarios son seudo nombre que puede ser usado para redireccionar correo a una cuenta de correo valida.Algunos usuarios tienen un alias de correo que apunta a un acronimo, titulo o nombre de departamento para asi poder crear una cuenta de negocio, para publicar en una página web o distribuir a cierta personas.

Ejemplo:
cc: root, miguel, carlos, jose
linux: ivelis, cesar, cristhian
info: root, pavel, carlos

Como podemos ver, la mayoría de los alias apuntan a un grupo de personas en vez de tener una cuenta que reciba los correos.

Las desventajas de los alias de e-mail ocurre cuando un empleado es dado entrada o es despedido. El administrador de sistemas debe recordar agregar un empleado al archivo de alias y eliminarlo si el empleado es eliminado. Un archivo de reenvío .forward no puede ser asociado a un nombre de alias. Los archivos .forward deben usar una dirección de correo valida. Los correos que son reenviados (forwarded) son una forma de alias. Un archivo .forward redirige cada correo y se lo envía a otra dirección de correo válida. A continuación presentamos la creación de un archivo .forward con un editor de texto como es el vi o nano:
pico .forward
desiree@abiertos.org
vi .forward
desiree@abiertos.org

Ya este es un archivo .forward válido. Recuerde que este archivo no direccionaría ningún archivo ya existente solo los nuevos. La gran mayoría de usuarios utilizan el archivo .forward to direccionar sus correos a una cuenta que ellos revisan regularmente.

El Archivo .vacation

El archivo .vacation está diseñado para que todos los correos entrantes sean autorespondidos, asi notificando al emisor que el receptor estará fuera de la oficina y pueda que no este revisando sus correos. El comando para invocar el archivo .vacation es vacation. El comando para detener el archivo .vacation es rm .vacation. El archivo .vacation.msg puede ser modificado para desplegar lo que realmente uno quiere que vean los que nos envían correos. Un listado de todas las personas que nos enviaron mensajes se guarda en el archivo .vation.db para referencia futura. El dueño de la cuenta con el archivo .vacation recibirá un correo del usuario para asi que el correo no rebote al emisor.

El archivo claro está, es oculto asi que escribir "ls -l" para mostrar el contenido del directorio no nos mostrará el archivo .vacation.msg, asi que tendrá que escribir el comando "ls -la" para poder ver el archivo. Y deberá usar el editor vi (o cualquier otro que edite texto simple) para editarlo.

Ejemplo

Este es un ejemplo del archivo .vacation.msg:

Subject: No estoy en la oficina por las próximas dos semanas. Su correo sobre $SUBJECT lo leere tan pronto regrese.

Sendmail.mc

El proceso de configuración de sendmail empieza con la creación de sentencia macro, la cual es usada para crear el archivo /etc/sendmail.cf. El sendmail.mc es creado por el macros m4. Después de haber creado el archivo sendmail.mc, los requerimientos mínimos son los parámetros Ostype y el Mailer. El Dominio (domain) y las Features (características) también son recomendadas. Ostype es el sistema operativo. Luego el Mailes debe ser todos los agentes de la entrega del correo como son local, mail, pop, smtp, usnet, uucp y fax. El dominio debe desplegar el dominio del cual usted es el host. Para terminar, las Features (características) son muy amplias pero muy necesarias para mostrar palabras claves.

Ejemplos de palabras claves incluyen allmasquerade, always_add_domain, y redirect. El archivo sendmail.mc contiene varios macros, algunos de los cuáles son incluidos para prevenir spam de email y comentar cierta líneas como la de dnl.

El archivo de macros de sendmail es algo asi:

m4 /etc/sendmail.mc > /etc/sendmail.cd

El procesor m4 se ejecuta con un macro y redirecciona la salida al archivo /etc/sendmail.cf

/etc/mail/xxx.db

En este directorio de correo hay muchas base de datos. Las base de dtaos están incluidas en el script makefile. Antes de editar el makefile, debe descargar la última versión de db. El makefile esta diseñado para especificar donde se encuentra su db. La db por lo general se encuentra en /usr o en /usr/local.

Listas de Correo

Las listas de correo pueden ser creadas usando el método de alias en dos maneras diferentes. En un caso, la dirección de la lista de correo puede ser ingresada en el archivo alias asi:

alias: receptores

Aquí, receptores es una lista de direcciones separadas por comas. Esta lista de receptores puede estar incluida en otros archivos y la podemos referir así:

:include:nombre_archivo

En este caso, las direcciones el nombre_archivo son consideradas como los receptores. Toda lista de correo debe tener un nombre de usuario al cual reportar todos los errores.

Comando newaliases

Sendmail no puede leer la versión texto del archivo aliases. Este archivo debe ser convertido en un formato binario ejecutando el comando newaliases.Cuando se efectúen cambios a este archivo /etc/aliases, el comando /usr/bin/newaliases debe ser ejecutado para reconstruir la base de datos.

Ejecutar sendmail

Una vez configurado sendmail, esta ya entonces puede ser ejecutado. Hay varias maneras diferente de completar esta tarea. La tarea más básica que sendmail puede ejecutar es la de enviar un correo electrónico. Para efectuar esto, ejecute la siguiente sentencia desde el directorio /usr/lib:

./sendmail <dirección e-mail> (Ej., ./sendmail info@abiertos.org)

Después de ejecutado el comando, entrará en un shell, donde se le permitirá digitar el cuerpo del mensaje. Para denotarle al shell que terminó debe digitarle un único punto en una línea nueva y luego presionar ENTER. Esto causará que su mensaje sea enviado a la dirección de correo especificada.

Sendmail también puede ser usado como un daemon de servicios, que envía los mensajes tomados desde una pila (qeue) cada x tiempo. Para correr sendmail como un daemon, ejecutelo con las siguientes opciones (flags):

./sendmail -bd -q3m

La opción -bd causa que sendmail se ejecute en el background como un daemon. La opción -q3m causa que el daemon sendmail envíe correos que se encuentran en la pila cada 3 minutos. Sendmail no coloca los mensajes de correo en la pila, ese trabajo lo efectúa el MUA.

Hay otras opciones muy útiles que pueden ser usadas con sendmail, y entre ellas se incluyen:

[-bi]	Usa la base de datos de alias definida en el archivo aliases
[-bp]	Imprime cual es la pila (qeue) de correo
[-bD]	Causa que sendmail se ejecute en el primer plano o foreground

Para aprender más sobre sendmail y los sistemas de correo puedes referirte a las secciones de nuestro portal donde tenemos una buena colección de How-Tos sobre sendmail y demá servicios que se pueden ejecutar sobre los sistemas GNU/Linux.

Diagnosticando Errores de Sendmail

En esta sección discutiremos algunos de los errrores y problemas más comunes que ocurren con sendmail. Incluido con cada discusión de un problema son las posibles causas y soluciones al problema.

El Error de “Relaying Denied”

Si al enviar correo a traves de sendmail ocurre el error de “Relaying Denied”, es importante investigar el porque. Como las restricciones por defecto de sendmail del relevo de correo, la configuración de sendmail debe ser cambiada para permitir el relevo (relay) de los mensajes. Hay tres maneras de lograr este objetivo con resultados diferentes de los noveles de control.

Agregar los Hosts al conjunto R

La clase del conjunto R es el conjunto de dominios permitidos a hacer relevo en sendmail. Una manera de resolver este problema es agregar cada nombre de host y su dirección IP que recibirá o enviará correo a la

clase del conjunto R.

Dependiendo de la versión de sendmail en uso, las especificaciones de la clase R se pueden encontrar en el archivo /etc/sendmail.cf para las versiones antes de la 8.8.x. Para las versiones 8.9.x en adelante en archivo se encuentra en /etc/mail/relay-domains.

Es importante agregar la dirección IP o nombre de host al cual el correo es enviado desde el conjunto R. Para asegurarse que la dirección IP será usada correctamente con cada DNS, deberá encerar la dirección [IP] entre llaves cuadradas.

Aplicar las características (FEATURES)

El método más preciso de controlar el relevo de correo dentro de sendmail es usar las características de sendmail. Las FEATURES aplicables a esta situación son relay_host_only. relay_entire_domain y relay_local_from. Para una más exhaustiva lista de características dirijase a las páginas Web de sendmail:

http://www.sendmail.org/tips/relaying.html

FEATURE(relay_hosts_only) es el equivalente de agregar un host al archivo /etc/sendmail.cf o a /etc/mail/relay-domains en la clase del conjunto R. FEATURE(relay_entire_domain) permite mandar y recibir mensajes desde cada dominio listado. Si el dominio es abiertos.org, agregar abiertos.org a este FEATURE permitirá enviar a y desde cualquier dirección en el dominico abiertos.org. La tercera característica, FEATURE(relay_local_from), permitirá relevo si el mensajes dice ser desde el dominio local. Claro esta, como esto es fácil de falsificar, esta FEATURE no es recomendada.

Modificar el Archivo de Acceso

Una tercera manera de controlar el relevo es usar el archivo de acceso de sendmail, el cual por lo general se encuentra en /etc/mail/access. Un control estricto puede ser obtenido si administrativamente colocamos un dominio en este archivo, en conjunto con especificaciones para aceptar o rechazar mensajes a o desde ese dominio. Los dominios listados pueden ser especificados con los formatos host.dominio.org, subdominio.dominio.org o dominio.org. Recuerde que si usamos FEATURES(relay_hosts_only), solamente la especificación host.dominio.org será aplicable en esta situación. En este siguiente ejemplo mostramos de un dominio que si aceptamos y uno que no aceptamos correo:

si_aceptamos.dominio1.org RELAY
rechazado.dominio2.org REJECT

Por lo menos uno de estos tres métodos deben resultarnos útil cuando enfrentamos un problema de el relevo de mensajes de un usuario a otro. Recuerde que sendmail deberá ser reiniciado después de efectuar uno de estos cambios, reiniciando el servicio o usando el comando SIGHUP.

¿Por Qué un Servidor No Se Nuevos Alias de Correo electrónico?

De vez en cuando al trabajar con un archivo alias de correo, enfrentaremos problemas donde nuestro correo no será correctamente entregado. Existen muchas razonas por la cual esto puede ocurrir.

Base de Datos de los Alias No Ha Sido Actualizada

Los cambios de los alias son efectuados en su archivo /etc/aliases. Para que eficazmente se encuentren los alias, estos deben ser insertados en la base de datos que se encuentra separada de este archivo. Este archivo de base de datos separado del /etc/aliases se encuentra en /etc/aliases.db. Cada vez que modificaciones son hechas al archivo de las alias, la base de datos deberá ser actualizada antes de que los cambios tomen efecto. Hay dos maneras de que los cambios se implementen. El primero es ejecutando el comando /usr/lib/sendmail -bi. El segundo es ejecutando el comando newaliases. Ambos comandos arrojan el mismo resultado- causar que la base de datos se reconstruya.

El archivo de los alias se lee una sola vez si sendmail se ejecuta como daemon. Asi que cualquier cambio que se efectúe a este no tomará efecto hasta que sendmail se reinicie. Pero aunque se reinicie el sendmail, si no reconstruimos la base de datos de alias entonces leería la información vieja.

Sintaxis Incorrecto

La configuración del archivo /etc/aliases requiere un sintaxis muy específico al efectuar cualquier modificación. Este sintaxis es:

alias: **destino**

El lado izquierdo define el nombre que el alias va a usar. El alias deberá ser seguido de inmediato por dos (:) puntos, luego unos cuantos espacios, y luego el destino para el alias. Asegúrese de no haber caracteres incorrectos o perdidos, ya que estos podrán causar problemas difíciles de solucionar. Aquí una lista de los caracteres especiales dentro del archivo de alias del sendmail:

#	Colocado al principio de una línea, anularía la línea por completo
/nombre_archivo	Colocado antes de la dirección destino, entregará la información al archivo
|programa	Colocado antes de la dirección destino, entregará la información a un programa
:include: lista	Usado para indicar que una lista de correo debe ser procesada

Cualquier error en el sintaxis de un alias en el archivo /etc/alisases causará que se altere la dirección de entrega del correo o que el correo sea devuelto. Asi que es de suma importancia asegurarse de un uso correcto de sintaxis al editar este archivo.

Alias Duplicado

Es muy posible en archivos de alias muy extensos que inserte un alias duplicado. Por ejemplo:

info: **ivelis**
info: **miguel**

En este caso, aparecería un mensaje de error al compilar o reconstruir la base de datos. Si uno quisiera que ambos ivelis y miguel sean referenciados por el alias info, la solución es cambiar estas entradas a:

info: **ivelis, miguel**

De no hacerlo asi el alias no será agregado a la base de datos, y no tomará efecto el cambio.

Alias en Bucle

Se da el caso que alias se referencia una a la otra y causan un bucle. Este es un ejemplo de dos alias causando el efecto de bucle entre ellas:

info: **ivelis**
ivelis: **info**

En sendmail el alias info se convertirá en no útil. Antes de hacer lo mismo con ivelis, sendmail descubre el bucle y procede a rebotar los correos dirigidos a ivelis. La solución es eliminar uno de los alias. Como los alias se pueden tornar un poco complejos [iede ser que la autodetección de los alias falle, en archivos alias con muchas entradas.

Alias Deshabilitado

Se puede dar el caso en el que sendmail se ejecute con la opción -n desde la línea de comandos. Esta opción deshabilita el uso de alias en sendmail, y por esto mail rebotará si no hacemos referencias a direcciones de correo válidas y no a alias.

También, como no hay un sitio establecido por defecto para el archivo alias que sendmail debe usar, usted deberá definir este sitio. Asegúrese que el sitio del archivo es el correcto. Si no lo es entonces sendmail simplemente operará bajo la asumpción de que alias está deshabilitado.

¿Por Qué Sendmail se Cuelga en el Arranque del Sistema?

El utilitario sendmail, al igual que otros utilitarios basados en redes, dependen de la presencia de una red para funcionar apropiadamente. La razón más común para que sendmail se cuelgue en el proceso de boot está muy ligado al funcionamiento de la red. Durante el inicio, el utilitario sendmail intenta resolver el nombre host del equipo local a una IP para asi poder enrutar correo correctamente. Si la red por alguna razón está abajo, esta simple petición será imposible, dejando el programa sendmail colgado en el intento de solucionar la IP del equipo local. Después del timeout (tiempo agostado) expiró, el sistema continuará su proceso de arranque, aunque una solución simple existe.

Solución

Fuera de asegurase de que una conexión de red al host este presente, la mejor manera de solucionar este problema es agregar una entrada para el nombre del host de la máquina en el archivo /etc/hosts asi:

```
127.0.0.1        localhost.localdomain
192.168.2.7      smith  smith.abiertos.org
```

Claro deberá substituir el nombre correcto de host en “smith” y proveer toda la demás información pertinente. Esto permitirá al programa sendmail continuar su inicio ya que la petición de nombre al archivo /etc/hosts no dependerá de una conexión de red. Si la red esta arriba, Asegúrese que el archivo /etc/resolv.conf contenga la información apropiada y específica de sus servidores de nombre (DNS).

Smail

Los programas de transporte de correo más usados en GNU/Linux son qmail, exim, Postfix, smail y sendmail. Ellos son también conocidos como MTAs, ambos soportan UUCP y SMTP. Histéricamente, sendmail es el más viejo, poderoso, complejo en su uso y el más usado en el ambiente ÓNIX tradicional. Por otro lado, smail es más simple de instalar y configurar y funciona mejor en sistemas con limitaciones de memoria. Nunca debe combinar el uso de smail con sendmail.

Por defecto, smail procesa y entrega todo el correo entrante inmediatamente. Si existe un tráfico alto relativo, el usuario puede tener a smail colectar todos los mensajes en una pila y luego procesarla a intervalos regulares. Si se elige el modo de pila (qeue), smail almacenará todo el correo en el directorio de los mensajes /var/spool/smail. No los procesará hasta que no sea ordenada de efectuarlo explícitamente. Para procesar este correo cada 10 minutos, por ejemplo, el usuario debe agregar la opción -p10m desde la línea de comandos.

Para ver la pila, use el comando:

```
$ mailq -v
```

Smail puede ser configurada para funcionar en ambiente tanto de LAN o en modo de UUCP.

Configurar Smail

El archivo de configuración de smail se llama config y reside en el directorio /usr/lib/smail. En algunas distribuciones, este puede estar en /etc/smail. Diferentes versiones de smail también instalan sus binarios en diferentes directorios: /usr/bin, /usr/lib o /usr/sbin. De acuerdo con el FSS (File System Standard), el lugar correcto debe ser /usr/sbin/sendmail.

Configurar en un LAN

Generalmente los host en una LAN intercambian correo usando SMTP, POP, IMAP o TCP/IP y una máquina es disenana a manejar conexiones UUCP externas. Es mejor mantener todos los buzones o mailboxes en un sólo sistema de archivos que se monte en todos los hosts vía NFS. Asi proveyendo acceso a los usuarios a sus correos sin importar en que equipo se sienten a trabajar en la LAN. Las direcciones de los

emisores deben ser independiente de el equipo que el usuario se encuentre ingresado al momento de escribir el mensaje. Por lo general el nombre del dominio es usado en la dirección de correo y no la dirección del host. Asi pués, la dirección del emisor se establece independiente de el equipo en el cual se escribe el mensaje.

Configuración del UUCP

El archivo de configuración es simple:

```
# nombres de dominios
visible_domain=
#
# nombre en correo saliente
visible_name=
#
# nombre uucp
#
# uucp_name=
#
# smarthost
smart_host=
```

La primera línea válida le informa a smail del dominio al cual el sitio pertenece.

La línea visible_name contiene un único FQDN (Fully Qualified Domain Name) para ser usado en todos los correos salientes. Este nombre es usado al general la dirección de correo de los correos salientes.

La línea uucp_name debe tener el mismo valor que la anterior de visible_name.

La última línea específica la ruta usada para el enrutamiento smart-host. El smail reenviará cualquier correo para direcciones remota al smart host. La ruta especificada en el atributo smart_host será usada como ruta al smart host.

Ejecutar El smail

Usted ouede correr smail como un proceso daemon por separado, o este puede ser establecido en inetd para que maneje el puerto de SMTP y que invoque smail. Dentro de la red TCP/IP, smail es usado con más frecuencia como un daemon por separado. Al momento de arranque (boot), este es iniciado por el script rc.inet2 y espera en el background o segundo plano por la conexión TCP en el puerto SMTP.

Para trabajar con inetd, una línea como la siguiente debe ser incluida en el archivo config:

```
smtp      stream  tcp    nowait      root    /usr/bin/smtpd   smtpd
```

donde smtpd debe ser un vinculo simbólico al archivo smail binario.

Algunos programas de correo retroalimentan mensajes, por defecto, para ser enviados por rmail o sendmail. Por esto es que deben haber dos vínculos simbólicos a smail antes de su inicio: /usr/bin/rmail y /usr/sbin/sendmail. Podemos crear estos vínculos asi:

```
# ln  -s  /usr/local/bin/smail        /usr/bin/rmail
# ln  -s  /usr/local/bin/smail        /usr/sbin/sendmail
```

Todo smail daemon coloca todos los mensajes de error en un archivo log. Estos archivos log se encuentran en el directorio /var/smail/log. Hay dos archivos log llamados logfile y paniclog que proveéis el administrador de sistema con la información si algo paso mal.

Qmail

Este instructivo de como instalar es tomado en su mayor parte de Diego Bravo Estrada “Guia Breve de Instalar Qmail”. Qmail es un MTA que proporciona el servicio de correo electrónico en sistemas

GNU/Linux y ÓNIX. Está pensado como un reemplazo más seguro para el clasico sendmail y pequeño smail. Las razones principales para usar Qmail y no Smail son: por seguridad y por comodidad. Sendmail y otros MTA's menos utilizados han sido una fuente constante de "bugs" y "vulnerabilidades" explotables por los crackers. Según algunos, esto se debe a que toda la complejidad del sistema de correo recae en un solo programa que necesariamente debe ejecutarse con privilegios de administrador. Qmail intenta paliar este problema mediante un esquema modular en el cual diversas etapas del procesamiento del correo electrónico son llevadas a cabo por distintos procesos que mayormente se ejecutan con un usuario no privilegiado.

Asimismo, la modularidad mencionada permite intercalar con facilidad filtros personalizados a criterio del administrador, cosa poco usual en los MTA's sencillos, y compleja en el MTA sendmail. Un caso muy útil correspondería a la instalación de programas antivirus, a fin de filtrar los mensajes maliciosos que pasan por el MTA y pueden afectar sistemas operativos debiles (como son todaslas versiones de los sistemas de Microsoft) que son vulnerables a la problemática de virus.

Además, qmail promueve (aunque no obliga) al uso del mailbox en el "home directory" a fin de evitar los riesgos que presenta el tradicional directorio de mailbox común; también promueve el uso de un formato distinto para este archivo, de modo tal que sea invulnerable a los fallos imprevistos que fácilmente corrompen el mailbox tradicional.

Instalar Qmail

Si Ud. descarga qmail del site oficial, probablemente tendrá que compilarlo. Para esto, su sistema deberá disponer del compilador de lenguaje C. Si Ud. no desea hacer esto, los desarrolladores también mantienen paquetes RPMs y DEBs (binarios.).

Prerequisitos para la Instalación de Qmail

Es muy recomendable (casi imprescindible) instalar el programa tcpserver. Si por algún motivo no se desea emplear tcpserver, se puede usar una combinación de inetd con los tcp_wrappers tal como recomienda la documentación que acompaña a qmail. Finalmente (como aporte inédito) aquí mostramos cómo trabajar con xinetd aunque la solución tiene visos de crack".

Otros requisitos evidentes corresponden a una conexión de red, y posiblemente la facultad de configurar el servidor de nombre (nameserver) si se está instalando por primera vez un mailserver para las estaciones de un dominio. Aquí asumiremos que el usuario está familiarizado con la arquitectura de Internet de correo electrónico y con el DNS.

Instalación de qmail

Lo primero que debemos hacer es crear el directorio de trabajo para qmail. La sugerencia de los creadores es el directorio /var/qmail (hay pocos motivos para cambiarlo.) Así que el administrador ejecutará algo como:

mkdir /var/qmail

Qmail requiere la creación de diversos usuarios para su correcta ejecución. Estos son: alias, qmaild, qmaill, qmailp, qmailq, qmailr, y qmails. Si por algún motivo no se puede emplear estos pseudo-usuarios (por ejemplo, si algún usuario real coincide con los mencionados), entonces se deberá especificar los nuevos valores en el archivo conf-users. Igualmente se requiere de la creación de dos grupos (especificados en el archivo conf-groups.)

Para crear los usuarios y grupos usar:

groupadd nofiles

```
# useradd -g nofiles -d /var/qmail/alias alias
# useradd -g nofiles -d /var/qmail qmaild
# useradd -g nofiles -d /var/qmail qmaill
# useradd -g nofiles -d /var/qmail qmailp
# groupadd qmail
# useradd -g qmail -d /var/qmail qmailq
# useradd -g qmail -d /var/qmail qmailr
# useradd -g qmail -d /var/qmail qmails
```

En este paso se generan los ejecutables de qmail y se prepara el directorio de trabajo de qmail:

```
# make setup check
```

Luego se deberá especificar el nombre de nuestro host (incluyendo el dominio completo) mediante el comando config-fast del siguiente modo:

```
# ./config-fast sede.abiertos.org
```

En qmail, el correo para los usuarios especiales postmaster, MAILER-DAEMON y root, es redirigido hacia el pseudo-usuario alias. Esto requiere de la existencia de ciertos archivos en el "home directory" del pseudo-usuario alias:

```
# (cd ~alias; touch .qmail-postmaster \
      .qmail-mailer-daemon .qmail-root)
# chmod 644 ~alias/.qmail*
```

El correo dirigido a los usuarios locales debe ser almacenado en algún archivo o directorio (el mailbox.) Esto normalmente NO lo realiza el MTA, sino que lo delega a un programa auxiliar. Sendmail normalmente emplea a procmail para este fin. A fin de realizar una rápida puesta a punto, mantendremos el uso de procmail (luego veremos el agente alternativo que proporciona qmail llamado qmail-local.) Para esto tan sólo es necesario efectuar el siguiente comando:

```
# cp /var/qmail/boot/proc /var/qmail/rc
```

Sin embargo, procmail en este caso será ejecutado mediante un usuario no privilegiado, por lo que es menester cambiar los permisos del directorio de los mailbox (que se mantendrá en /var/spool/mail.) Para RedHat:

```
# chmod 1777 /var/spool/mail
```

Nótese que qmail recomienda cambiar el mailbox; sin embargo, esto requiere algunos pasos adicionales que no veremos aquí. Lea la documentación respectiva (archivo INSTALL.mbox) para más detalles.

Lo primero que debemos hacer es asegurarnos de que sendmail no está en ejecución, por lo que habremos de buscarlo en la tabla de procesos:

```
# ps ax | grep sendmail
  698 ?      S    0:00 sendmail: accepting connections
  787 pts/0  S    0:00 grep sendmail
```

La primera línea indica que sendmail está en ejecución, por lo que debemos hacer que termine. Para esto usaremos cualquiera de los siguientes comandos (basta con uno) en orden de preferencia:

```
# service sendmail stop
# /etc/rc.d/init.d/sendmail stop
# kill 698
```

El “698” del último comando corresponde al PID del proceso y se obtiene del comando anterior. En su sistema deberá presentarse otro valor. Asegúrese (volviendo a lanzar “ps”) de que sendmail ya no esté en ejecución.

Luego debemos asegurarnos de que sendmail no se vuelva a ejecutar. La manera más sencilla consiste en desinstalarlo mediante:

rpm -e sendmail (también podía ser dpkg -e sendmail)

Probablemente deberá desinstalar otros paquetes (como fetchmail y mutt.) Para forzar la desinstalación de sendmail:

rpm -e --nodeps sendmail

El archivo REMOVE.sendmail muestra otras maneras de trabajar sin necesidad de eliminar el paquete sendmail, aunque lo anterior es más recomendable.

Diversos programas asumen la existencia de sendmail y lo invocan ciegamente. Por esto, qmail proporciona un “reemplazo” básico para sendmail, a fin de mantener operativas a las aplicaciones mencionadas.

ln -s /var/qmail/bin/sendmail /usr/lib/sendmail

ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail

Qmail proporciona páginas de manual para diversas utilidades. Estas se instalan en /var/qmail/man. Sin embargo, el sistema “man” debe ser configurado para acceder a éste.

Para esto, añada el directorio de los manuales mediante el la directiva MANPATH en el archivo /etc/man.config:

```
...
MANPATH /usr/share/man
MANPATH /usr/man
MANPATH /usr/X11R6/man
MANPATH /usr/lib/perl5/man
MANPATH /usr/kerberos/man
MANPATH /usr/local/man

# Añadido para qmail
MANPATH /var/qmail/man
. . .
```

Luego, Ud. deberá probar algo como man qmail-send.

Instalación de tcpserver

Qmail necesita de un mecanismo que lance el demonio qmail-smtpd cada vez que llega un intento de conexión SMTP del exterior del mailserver. Esto se puede hacer de diversas maneras; sin embargo, los creadores recomiendan el uso del programa tcpserver que está disponible como parte del paquete ucspi-tcp de D.J. Bernstein. Es posible configurar inetd para este fin (ver el archivo FAQ) e incluso xinetd como ilustro en esta misma guía en otra sección.

En el Ud. deberá encontrar la última versión del paquete. Este viene en un archivo TAR comprimido. En mi caso, ucspi-tcp-0.88.tar.gz. Ud. deberá luego desempacarlo en un lugar razonable (después lo podrá eliminar) mediente un comando como:

cd /usr/local

tar xvzf /ruta_al_empaquetado/ucspi-tcp-0.88.tar.gz
....
se crea el directorio ucspi-tcp-0.88.....

Ud. procederá ahora a compilar los programas del paquete. Para esto ejecute:
cd ucspi-tcp-0.88
make

Y tras unos momentos deberá tener una serie de ejecutables en el mismo directorio. Proceda a copiar los ejecutables tcpserver y tcprules a un directorio en el PATH, como /usr/sbin o /usr/local/bin:
cp tcpserver tcprules /usr/sbin

Sólo estos dos ejecutables serán necesarios, por lo que puede eliminar el directorio de ucspi-tcp-0.88.

Configurar el inicio automático de qmail

Ahora vamos a configurar el sistema para que siempre se ejecute qmail al reiniciarse el computador. Ud. deberá averiguar el UID y el GID del usuario "qmaild" y del grupo "nofiles" respectivamente:

id qmaild
uid=1003(qmaild) gid=1002(nofiles) groups=1002(nofiles)

El número 1003 es el UID del usuario "qmaild", y el número asociado al grupo "nofiles" es 1002. Es seguro que estos valores serán distintos en su sistema. Ahora, simplemente añada los siguientes comandos:
```
csh -cf '/var/qmail/rc &'
/usr/sbin/tcpserver -u 1003 -g 1002 0 smtp \
        /var/qmail/bin/qmail-smtpd &
```
al final del archivo /etc/rc.d/rc.local.

Probando qmail

Para las pruebas que siguen, se recomienda disponer de un computador auxiliar configurado para enviar y recibir correo (posiblemente usando sendmail!) En lo que sigue, denominaremos remoto.unima.edu.pe a este computador.

Qmail enviará mensajes de diagnósitco a syslog, por lo cual normalmente deberíamos buscarlos en el archivo /var/log/maillog que es donde syslog imprime los mensajes de email en RedHat (para más información, ver /etc/syslog.conf y el manual de syslog.conf)

Hemos configurado qmail para que se ejecute cada vez que el computador es reiniciado. Así que conviene en este momento reiniciar el computador a fin de apreciar que hemos realizado bien esta tarea.

Para analizar si los procesos de qmail están en ejecución, pruebe a lanzar el siguiente comando:
```
# ps axu | grep qmail
qmails    3727  1392    pts/2 S        Jan13  0:00 qmail-send
qmaill    3728  1360    pts/2 S        Jan13  0:00 splogger qmail
root      3729  1348    pts/2 S        Jan13  0:00 qmail-lspawn
qmailr    3730  1348    pts/2 S        Jan13  0:00 qmail-rspawn
qmailq    3731  1340    372 pts/2 S    Jan13  0:00 qmail-clean
```

Casi todos los valores numéricos serán distintos en su computador (incluso hemos recortado un poco la salida por cuestiones de formato.) Lo importante es el nombre de los procesos, y de los usuarios dueños de los mismos (a la extrema derecha e izquierda, respectivamente.) Revise el "log" si no se hubieran iniciado

estos procesos.

En caso de que no pudiera reiniciar el computador en este momento, tendrá que iniciar qmail manualmente:

csh -cf '/var/qmail/rc &'
/usr/sbin/tcpserver -u 1003 -g 1002 0 smtp /var/qmail/bin/qmail-smtpd &

Se trata de observar si los mensajes de qmail pueden ser distribuidos, es decir, pueden ser enviados a los usuarios de nuestro computador o de otro.

En primer lugar, "inyectamos" un mensaje con destino local, para lo cual Ud. deberá modificar la palabra usuario por el nombre de un usuario común del sistema (distinto del administrador.) Esta prueba la debería efectuar con un usuario común.

$ echo to: usuario | /var/qmail/bin/qmail-inject

Escriba con cuidado el "to:" separando el nombre de usuario. Observe el log, y pruebe a recepcionar este mensaje (por ejemplo, logueandose con el usuario y ejecutando $ mail.) Pruebe también el envío a un usuario local inexistente:

$ echo to: inexistente | /var/qmail/bin/qmail-inject

Pruebe ahora el envío a un computador remoto:

$ echo to: usuario@remoto.abiertos.org | /var/qmail/bin/qmail-inject

Esto requiere que qmail se conecte al puerto SMTP de "remoto" (donde también deberá haber un servidor de email) y que allí exista el usuario especificado.

El archivo TEST.deliver contiene más pruebas y detalles que Ud. debe leer.

Ahora se trata de observar si qmail acepta mensajes (que después serán distribuidos como se vió arriba.) Para esto, tcpserver debe estar "escuchando" en el puerto SMTP (25). Esto puede analizarse fácilmente con netstat:

netstat -a —inet | grep smtp
tcp 0 0 *:smtp *:* LISTEN

Si esto tarda mucho, o no funciona bien, usar:

netstat -an —inet | grep 25
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN

Desde el computador remoto.unima.edu.pe, pruebe a enviar un mensaje a a cualquier usuario (no root) del computador local marquesa.unima.edu.pe, y observe los mensajes del log de ambos sistemas.

El archivo TEST.receive contiene más pruebas y detalles.

Qpopper

Qpopper es un programa, compatible con servidores POP3, que permite la descarga de correos electrónicos desde programas clientes. Este se ejecuta en un host conectado al Internet y permite que los correos sean leídos sin conexión. Qpopper permite que los usuarios que ejecutan clientes POP en sus equipos conectarse a la LAN, conectarse al servidor POP y descargar los mensajes de correo a su equipo cliente. Este no tiene su propio MTA y ni tampoco proveéis soporte SMTP. Trabaja usando agentes estándares de transferencia ÓNIX como son sendmail y smail. Este servidor ya cumple completamente con el soporte para POP y soporta todos los clientes POP3. Su última versión es la 4.0 y puede ser descargada desde su ftp. Qpopper sopor-

ta protocolos de autenticación segura como usuario/contraseña, autenticación POP (APOP) y Kerberos. También puede ser usado con PAMs (Plugable Autentification Modules).

Basta con instalar el paquete qpopper para comenzar a utilizar el servicio. Pero hay que tener en cuenta un par de factores para que funcione bien y en forma segura. qpopper se ejecuta como un servicio inetd, o sea que se agrega (durante la instalación) una línea en nuestro /etc/inetd.conf como las siguiente:

```
#:MAIL: Mail, news and uucp services.
 pop-3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.qpopper -f /etc/qpopper.conf
```

Si utilizamos el servicio portmap (y sería razonable hacerlo) debemos configurar debidamente los archivos de configuración /etc/hosts.deny y /etc/hosts.allow. Un esquema de seguridad lo más restrictivas posibles, puede ser como este ejemplo del archivo hosts.deny es asi:

ALL: ALL

De esta manera ningún servicio de los que controla inetd es aceptado, excepto los establecidos en /etc/hosts.allow. Para que sea aceptado un pedido al servicio qpopper debe existir la siguiente línea en ese archivo:

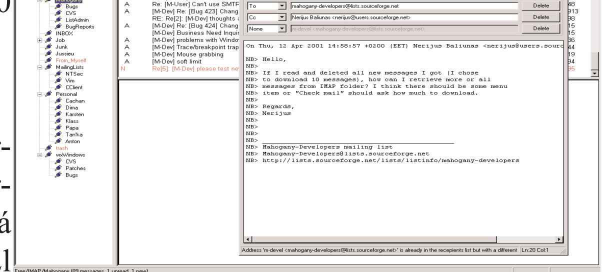
in.qpopper : 192.168.95.0/24 :spawn (echo "c(%c) d(%d)" >> /tmp/tcpw.log)

En cuyo caso sólo se aceptarán pedidos que provengan desde mi red privada, y rechazo lo que venga desde cualquier otra dirección.

El otro punto fundamental para asegurar nuestro servicio qpopper es aplicar las reglas correctas en el lugar apropiado de nuestro archivo de configuración de firewall, en este caso iptables. Abajo detallo las líneas necesarias para permitir paquetes al puerto 110 de pop3, con política DROP por default a todos los paquetes.

Permitimos paquetes contra el puerto 110 para correo pop3
iptables -A INPUT -i $LAN_LOCAL -p tcp —sport 1024:65535 —dport 110 -j ACCEPT
iptables -A OUTPUT -o $LAN_LOCAL -p tcp —sport 110 —dport 1024:65535 -j ACCEPT

Lo que hemos hecho fué aceptar pedidos a través de nuestra tarjeta de red local de paquetes entrantes desde en un rango de puertos no privilegiados, de protocolo tcp contra nuestro puerto 110, y dejar salir por la misma interfaz y el mismo protocolo paquetes originados en nuestro puerto 110 contra los no privilegiados de la otra máquina.



Mahogany

Mahogany es un cliente de correo y noticias opensource. Es un programa de plataforma cruzada que soporta programas servidores de SMTP, POP3, IMAP y NNTPy está disponible tanto para X11/ÓNIX, MacOS y MS-Windows. El cliente de correo y noticia es capaz de recibir y almacenar noticias y mensajes de correo desde servidores compatibles. Mahogany fué desarrollado usando la arquitectura X Window, construida sobre el toolkit de GTK sobre ÓNIX. Las ultimas versiones tienen soporte completo para MIME, edición de MIME en la ventana de composición y soporta para visores externos. Soporte para click sobre direcciones URL y la recepción completa de faxes es incluida. Soporte internacional completo, una interfase GUI amistosa, y de muy fácil uso. Soporte para sincronización de PalmOS, soporte de SSL, soporte de dial up.

Clientes de Correo de Texto Plano

La mayoría de los clientes de correo electrónico modernos permiten al usuario seleccionar si desea enviar los correos en formato de texto plano (sin formato) o en formato HTML. La ventaja del correo electrónico con formato HTML es que pueden contener gráficos y enlace interactivos para los sitios Web. Se puede especificar el tipo de fuente concreto, el diseño es muy cómodo y se pueden agregar fondos, imágenes y texturas; todo esto aporta al mensaje un aspecto muy atrayente para el destinatario.

Por otro lado, un mensaje de correo electrónico en texto plano es simplemente eso, texto plano sin formato. No son elaborados ni tiene imágenes incrustadas en el correo electrónico. Tampoco tienen tipos de letra especiales. Los mensajes de texto plano son sencillos.

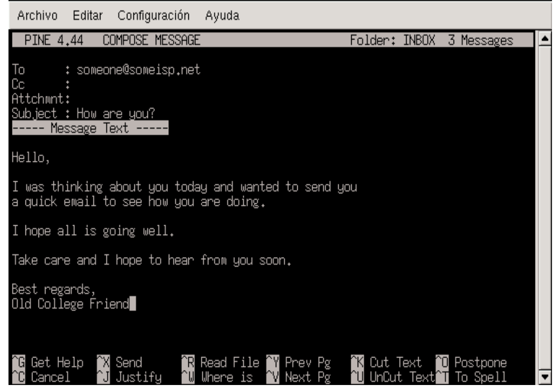
El término “texto plano” hace referencia a datos de texto con el formato ASCII. El texto plano (también denominado texto sin formato) es el formato más portátil porque es compatible con casi todas las aplicaciones de varios tipos de máquinas.

En esta sección se describen dos clientes de correo electrónico de texto plano, Pine y mutt.

El MUA Pine

Pine and mutt (el acrónimo de pine es elm o Program for Internet News and Email) es un cliente de correo electrónico basado en caracteres para sistemas ÓNIX. Para lanzar Pine escriba pine como comando en el indicador de comandos de la shell. Una vez iniciado pine, aparecerá la pantalla Main Menu.

Todas las pantallas de pine tienen un diseño similar: la línea superior indica el nombre de la pantalla e información adicional útil, debajo se encuentra el área de trabajo (en la pantalla Main Menu, el área de trabajo es un menú de opciones), a continuación figura la línea de mensajes y el indicador de comandos y, por último, está el menú de comandos.



En Main Menu, puede seleccionar opciones para leer la ayuda en línea, componer y enviar un mensaje, mirar un índice de mensajes de correo, abrir o mantener las carpetas de correo, actualizar la libreta de direcciones, configurar pine y salir de pine. También hay opciones adicionales en la parte inferior de la pantalla.

Para escribir un mensaje, presione [C] (abreviación de Compose). Aparecerá la pantalla Compose Message. Vea la pantalla de composición de mensajes de Pine

Según si el cursor se sitúa sobre un campo u otro de la pantalla, se presentarán diferentes comandos. Para ver otros comandos disponibles al situar el cursor sobre el campo Message Text, presione [Ctrl]-[G] (Obtener ayuda). Por ejemplo, para desplazarse, utilice las teclas de flecha o [Ctrl]-[N] (Línea siguiente) y [Ctrl]-[P] (Línea anterior); para corregir los errores tipográficos, utilice las teclas [Retroceso] o [Supr].

En el menú de comandos anterior mostrado en Figura, se ha utilizado el carácter ^ para indicar que se trata de una tecla de control. Este carácter significa que debe mantenerse presionada la tecla Control ([Ctrl]) simultáneamente a la letra correspondiente de cada comando. Si desea salir de Pine, presione [Q] (Salir).

Para ver un mensaje en la pantalla Message Index, utilice las teclas de flecha para resaltar el mensaje que desee ver. Presione [V] (Ver mensaje) o [Intro] para leer un mensaje seleccionado. Para ver el siguiente mensaje, presione [N] (Siguiente mensaje). Para ver el mensaje anterior, presione [P] (Mensaje anterior). Para volver del mensaje al índice de mensajes, presione [I] (Índice).

Para obtener información adicional sobre pine, consulte la página del manual de pine. Para ver esta página del manual, escriba el comando man pine en el indicador de comandos de shell.

El MUA mutt

Mutt es un cliente de correo basado en texto de reducido tamaño pero muy eficaz para sistemas operativos ÓNIX. El archivo de configuración de Mutt, ~/.muttrc., confiere a mutt una gran flexibilidad y capacidad de configuración. También es este archivo el que puede ocasionar problemas a los nuevos usuarios. El número de opciones que mutt tiene disponibles es verdaderamente sorprendente. mutt permite al usuario controlar todas las funciones que mutt utiliza para enviar, recibir y leer el correo. Al igual que sucede con cualquier otro tipo de software de estas características, lleva mucho tiempo comprender las funciones y dominar lo que se puede hacer con ellas.

La mayoría de las opciones se llaman utilizando los comandos set o unset, bien con valores booleanos o de cadena. Por ejemplo, set folder = ~/Mail.

Todas las opciones de configuración se pueden cambiar en cualquier momento si se escriben dos puntos ([:]) seguidos del comando correspondiente. Por ejemplo, :unset help desactiva las útiles sugerencias del comando del teclado en la parte superior de la pantalla. Para volver a activar estas sugerencias, presione :set help.

Si no puede recordar el comando que desea utilizar, siempre podrá utilizar una función para rellenar la ficha que le será de ayuda.

No tiene que escribir todos los comandos de configuración preferidos cada vez que ejecute mutt. Puede guardarlos en un archivo que se cargue en el momento de iniciar el programa. Este archivo de configuración debe guardarse en el directorio principal y se tiene que denominar ~/.muttrc o ~/.mutt/muttrc.

Al lanzar mutt, lo primero que verá es una pantalla con una lista de los mensajes de correo electrónico. Este menú inicial se denomina índice. Ver la Figura pantalla principal de mutt.

Estos mensajes se almacenan en una carpeta de correo predeterminada que se suele denominar mailspool, lo que sugiere la idea de una bandeja de entrada. Utilice las teclas [K] y [J] del teclado para mover el cursor hacia arriba y hacia abajo por la lista de mensajes

En las vistas de índice o localizador, utilice la tecla [R] para contestar a un mensaje o la tecla [M] para crear uno nuevo. Mutt le pedirá que defina la dirección To: y la línea Subject: . A continuación, se lanzará un editor de texto (definido por la variable de entorno $EDITOR en el archivo de configuración) con el que podrá componer el mensaje.

Después de editar su correo electrónico, Mutt abrirá el menú de composición, donde puede ajustar las cabeceras de los mensajes, cambiar la codificación, agregar archivos adjuntos o, simplemente, presionar la

tecla [Y] que significa sí para enviar el correo electrónico.

Para obtener más información sobre Mutt, consulte las páginas del manual de muttrc y mutt (escriba man muttrc o man mutt en el indicador de comandos de shell). También puede consultar el manual mutt que será muy útil. El manual mutt se instala en /usr/share/doc/mutt-1.2.x, donde x es el número de versión del programa mutt instalado en el sistema.

E-MAIL VIRTUAL

El e-mail virtual permite que usuarios emplen un alias de correo en un dominio en particular y que ese correo sea automáticamente reenviado a otra dirección en un dominio diferente. Existen varias configuraciones y aplicaciones al correo virtual. Usando correo virtual, todo el correo de un dominio puede ser reenviado a un solo buzón, alias en particular en un dominio pueden ser reenviados a buzones individuales y los alias no mapeados en un dominio pueden ser enviados a una cuenta general de correo. Por ejemplo, el correo de info@abiertos.org y soporte@abiertos.org pueden ser automáticamente redireccionados a una cuenta de un usuario como por ejemplo j_sanchez@otrodominio.org. Claro está, correo enviado a una cuenta o alias en un servidor virtual que no existen pueden ser enviadas a una cuenta general. Correo enviado a desiree@abiertos.org donde esta cuenta o alias no existe sea direccionada a un destino predefinido.

Hay que tomar unas cuantas consideraciones especiales al establecer cuentas de correo virtuales. En un servidor virtual, uno puede asociar multiple nombres de dominios con ese servidor. Cualquier cuenta de correo o alias que existe en ese servidor puede ser asociados con cada uno de los nombres de dominios. En la situación donde un servidor virtual soporta los nombres de dominios dominio1.com y dominio2.com, y una cuenta de correo ya existente c_brito, entonces las direcciones c_brito@dominio1.com y c_brito@dominio2.com son ambas válidas. Normalmente, correo enviado a una de esta cuenta entonces fuera redireccionada a una cuenta única; pero, es posible tener correo para c_brito@dominio1.com y c_brito@dominio2.com redireccionado a direcciones diferentes usando mapeo virtual (virtmaps). Un virtmap es una función como los alias del e-mail en que esta automáticamente re-enruta los correos. Donde alias de emails son almacenadas en el archivo /etc/mail/aliases, virtmaps son almacenados en el archivo /etc/mail/virtmaps. Diferente a alias de e-mails, virtmaps sólo pueden apuntar un e-mail a una dirección de destino. También, los virtmaps requieren la dirección completa de correo no solo una cuenta o alias.

La configuración del archivo virtmaps es relativamente fácil. Lo primero es abrir el archivo en un editor de texto. Si este archivo no existe, debe actualizar la configuración de correo con el comando updatesendmailcf. Ahora, editar el archivo virtmaps, escriba la información en un formato de dirección de entrada/salida. Por ejemplo:

cuenta_nombre@dominio.com destino

En este ejemplo, cuenta_nombre puede ser cualquier nombre, y dominio.com representa el nombre de un dominio. El destino puede ser cualquier alias, cuenta de correo, virtmap en un servidor virtual, o cualquier dirección de e-mail para una cuenta que no existe en un servidor virtual. Aquí dos ejemplos más:

jazmine@dominio.com jazzy@abiertos.org
desiree@dominio.com dessi@abiertos.org

En la primera línea jazmine@dominio.com será direccionada a jazzy@abiertos.org. De la misma forma desiree@dominio.com será reenviada a jazzy@abiertos.org. Después de establecer el archivo virtmaps, es importante ejecutar el comando vnewvirtmaps para implementar el nuevo virtmap.

Podemos también establecer un virtmap que direccionará todo el correo enviado a un dominio con cuentas o alias no válidas o sin cuenta o alias no especificado a una cuenta por defecto en vez de retornarlas al

emisor. Tal virtmap es llamado un aparalo todo (catch-all). Un virtmap catch-all toma todo correo enviado a cuentas no existentes y no válidas y las envía a una cuenta por defecto. El archivo para el virtmap catch-all está almacenado en el directorio /etc/mail, y el sintaxis es igual que el virtmap normal. La diferencia es que el primer elemento es el nombre de dominio sin cuenta de correo o sin especificar un alias. Por ejemplo:

abiertos.org cuenta_general@abiertos.org

Esta línea redireccionara cualquier correo dirigido a abiertos.org que no esta ya mapeado a una cuenta de correo destinataria a una cuenta general cuenta_general@abiertos.org. Como con cualquier otro virtmap, debemos actualizar la base de datos antes de que el virtmap catch-all funcione. Esto se efectúa con el comando vnewcatchall.

El sendmail utiliza otro método muy similar para configurar email virtual. Este utiliza el archivo virtusertable localizado en /etc/mail que es donde se le específica redireccionar el correo. El sintaxis de la información en el archivo virusertable es idéntica a la del archivo virtmap. La información de redirección es insertada en el formato de dirección/destino. Diferente a los virtmaps, la información de e-mail virtual y catch-all es contenida toda en un mismo archivo. La línea de catch-all es colocada al final del archivo virtusertable. Sendmail lee el archivo una vez, y si la dirección que busca no es encontrada, entonces asumirá que el nombre de cuenta o alias que busca no es valida y la enviará a la dirección de destino catch-all. Otra diferencia muy importante con el sendmail es que el usuario deberá incluir el símbolo de @ antes del nombre del dominio en la sentencia catch-all. Por ejemplo, use @dominio.org y no sólo dominio.org

Aún otra manera de configurar la entrega de e-mail virtual es a traves de la modificación del archivo .procmailrc en el directorio home del usuario. Si usa solo una cuenta de correo virtual y desea tener el correo redireccionado a otra cuenta de correo en otro dominio, la información debe ser colocada ene el archivo .procmailrc en el siguiente formato:

```
:0
! j_paredes@abiertos.org
```

j_paredes@abiertos.org se convierte en la cuenta de correo deseada para reenviar el correo. En una situación donde usted desea direccionar más de un alias de e-mail a direcciones diferentes, use el sintaxis aqui detallado:

```
:0
* ^TO_peque@abiertos.org
! j_paredes@abiertos.org

:0
* ^TO_alex@abiertos.org
! j_paredes@abiertos.org
```

Estas entradas entregaran correo de peque@abiertos.org a j_paredes@abiertos.org y el correo de alex@abiertos.org a j_paredes@abiertos.org.

SERVIDORES DE LISTAS

Mantener una lista de correo (mailing list) y enfrentar las necesidades constantes de los usuarios ocupa mucho tiempo y produce trabajo muy tedioso. Subscribir y eliminar usuarios del sistema puede llevarse más tiempo que el de escribir la carta de socios (newsletter). Las listas de correo basicamente funcionan estableciendo una cuenta que redirecciona todos los correos que recibe a un número de gente contenidos en la lista. ¿Qué sucede si el dueño de la lista desea editar los correos antes de reenviarlos? ¿Qué sucede si desea crear directorios para guardar mensajes para luego ser accesados remotamente? ¿Cómo pode-

mos automatizar tareas simple que deben ser ejecutadas periódicamente? Estas son algunas de las razones por las cuáles existen los MLMs (Mail List Managers) como son Mailman, Listserv, Majordomo, y SmartList. Ellas administran subscripciones, moderan mensajes, postean y detectan/eliminan direcciones erroneas. Cada uno de estos paquetes ofrecen variadas características. Indiferentemente si su lista es local o remota un buen entendimiento de su paquete MLM es muy importante para la administración de us lista.

Instalar y configurar los diferentes MLMs difiere en algunos aspectos, y algunas puede ser más´s difíciles que otras. Algunos MLMs requieren el paquete ProcMail, el cual se ejecuta en conjunto con Smartlist cual debe ejecutarse simultáneamente. MLMs no pueden enviar sus propios correos y necesitan un programa de correo. Todos ellos pueden usar sendmail, el cual viene con las distribución estandar de GNU/Linux, y muchos pueden usar sistemas de correo de versiones comercial.

En esta sección discutiremos los siguientes tópicos:
- Mailman
- Majordomo
- Procmail
- LISTSERV

Listas de correo con Mailman

Vamos a ver como instalar nuestro servidor de listas de correo, paso a paso:
En primer lugar, descargamos el código fuente, del ftp oficial:
•wget http://ftp.gnu.org/gnu/mailman/mailman-2.1.3.tgz

Lo descomprimimos en /usr/src
• tar xvzf mailman-2.1.3.tgz -C /usr/src

Nos posicionamos en el directorio:
• cd /usr/src/mailman-2.1.3

Ahora vamos a crear el usuario y grupo mailman, de sistema, necesarios para funcionar:
• adduser —system —no-create-home —disabled-password —disabled-login mailman

Antes de compilar, crearemos el directorio donde vamos a instalar, y le damos los permisos adecuados, con los comandos:
•mkdir /home/mailman-2.1.3
•chgrp mailman /home/mailman-2.1.3
•chmod a+rx,g+ws /home/mailman-2.1.3

Preparamos la instalación con el script configure:
./configure —prefix=/home/mailman-2.1.3/ —exec-prefix=/home/mailman-2.1.3/ —with-mail-gid=vchkpw —with-mailhost=debianitas.net —with-urlhost=mail.debianitas.net

Un poco más abajo tienes información sobre la ejecución del configure.
Una vez que ya tenemos todo listo, instalamos con:
• make install

Comprobamos que los permisos son correctos con el script:
• bin/check_perms

Como seguramente no lo sean, los ponemos bien añadiendo un -f a dicho script.

• **bin/check_perms -f**

Por ultimo , crearemos un enlace genérico que nos servirá para cuando cambiemos de versión.

• **ln -s /home/mailman-2.1.3 /home/mailman**

Con esto, ya tendremos instalado nuestro servidor de listas de correo! Pero antes de que te apresures, tenemos que configurarlo de forma adecuada.Dependiendo del SMTP (servidor de correo) que tengamos, la configuración de las listas de correo variara. Esto se debe a que mailman trabaja principalmente con alias de correo (direcciones de correo que no se corresponden a usuarios reales), y a que los alias de correo varían dependiendo del servidor de correo que estemos utilizando.

En mi caso concreto, para el mantenimiento de diversos dominios de correos utilizo qmail junto con vpopmail, lo que complica aun más la situación.

A modo de breve indicación, en qmail los alias son archivos de correo, del tipo .qmail-macklus (un alias tal que macklus@dominio.net), que contienen la acción que se ha de tomar al recibir un correo (redirigirlo a otra cuenta, procesarlo a través de un programa externo, etc). Además, con vpopmail tenemos varios dominios, situados todos ellos en /home/vpopmail/dominio.ext , lo que significa que para nuestra configuración necesitamos que los alias se creen en el directorio correcto (en el /home/vpopmail/dominio.ext anteriormente indicado, pero con el mismo formato que se usa en qmail).

Revisando la documentación, encontré el script qmail-to-mailman.py , ofrecido por el propio mailman, pero que no se adapta a lo que yo deseaba. Para solventar el problema, basto crear un script en sh que:

• Comprueba que los directorios necesarios existen.
• Permite utilizarlo con cualquier dominio que tengamos en vpopmail.
• Crea los alias de forma automática.
• Crea los parámetros de la lista en el propio mailman.

Este script se encuentra en el anexo que aparece a continuación.

Para cualquier otro servidor de correo, en principio podrás utilizar los alias que aparecen cuando creas una lista con newlist, debido a que la mayor parte de los servidores de correo utilizan el mismo sistema.

Los siguientes enlaces pueden serte de gran ayuda:

• Pagina web de mailman:
• Apache, el mejor servidor de páginas web
• Documentación sobre apache y cgi-bin (versión 2 de apache):

Listas de correo con Majordomo

Majordomo es una de los programas MLM más usados. Una vez instalado, este es operado remotamente por correo por ambos usuarios (los cuáles pueden subscribirse y cancelar desde la lista sin la

intervención del mantenedor) y administradores (pueden efectuar todas las funciones de mantenimiento remotamente). Majordomo no envía correo por si misma. Es basicamente una colección de archivos scripts escritos en Perl que controlan una lista de direcciones de parte de un MTA por separado como es el configurado por defecto de majordomo sendmail.

Majordomo es un conjunto de programas que permiten administrar eficientemente las listas de correo electrónico en Internet, ya que reduce al mínimo la intervención del administrador.

Para los usuarios, majordomo es un domicilio electrónico al que se le envían todas las preguntas referentes a las listas de correo que administra el programa. Si un usuario desea ser miembro de alguna de estas listas, recurre a majordomo para que le dé toda la información necesaria.

Mediante este programa se solicita el registro y la suscripción a un foro de discusión, y una vez que se es miembro de éste se puede participar en él. Se puede saber qué listas serán atendidas por majordomo, inscribirse a ellas, preguntar quienes están en éstas con sólo enviar un comando al servidor de majordomo (majordomo@hostname). Este entenderá los comandos que le lleguen por correo.

Majordomo puede ser configurado para soportar diferentes tipos de listas. Una de las razones de su popularidad es por su tamaño reducido, gratuita y escrito en Perl, lo cual lo hace relativamente fácil de entender y de individualizar. Su instalación no es difícil y sus características de uso es simple. Interfases vía web están disponible para ambos el usuario y la administración.

Como comunicarse con Majordomo

Para efectuar las peticiones antes mencionadas se debe enviar un mensaje al servidor de majordomo.

majordomo@nombre_de_host

donde nombre_del_host es el nombre de la máquina donde reside Majordomo. Se puede enviar a Majordomo una petición a través de un comando por correo electrónico. Estos comandos se envían en el cuerpo del mensaje.

Procmail

Procmail le permite filtrar el correo a medida que lo recibe de un servidor de correo remoto o colocarlo en el archivo spool de un servidor de correo local o remoto. Es una herramienta eficaz, que hace un uso adecuado de los recursos del sistema y de amplio uso. Procmail, comúnmente denominado LDA (Local Delivery Agent. Agente de entrega local), desempeña una pequeña función en la entrega de correo para su lectura por un agente MUA.

Para utilizarlo, primero deberá instalarlo. Escriba el comando rpm -q procmail para ver si tiene instalado el paquete procmail. Si, por alguna razón, Procmail no está en el sistema, instálelo utilizando los CD-ROM de instalación de Red Hat Linux.

Procmail se puede llamar de distintos modos. Puesto que el correo se ubica en el archivo spool de correo electrónico, Procmail se puede configurar para arrancar, filtrar el correo en las ubicaciones configuradas para utilizar con el agente MUA y salir. Opcionalmente, puede configurar el agente MUA para iniciar Procmail en cualquier momento que se reciba un mensaje para que los mensajes se trasladen a los buzones correctos. En muchos casos, la presencia de un archivo .procmailrc en el directorio principal del usuario llamará la actividad de Procmail, si se utiliza Sendmail.

Las acciones que Procmail realiza con el correo dependen de las instrucciones de las recetas o reglas mediante las que se comparan los mensajes con el programa. Si un mensaje coincide con la receta o regla, el

correo se ubicará en un determinado archivo, se eliminará o se procesará.

Cuando Procmail arranca, lee el mensaje de correo y separa el cuerpo de la información de cabecera. A continuación, busca el archivo /etc/procmailrc y los archivos rc en el directorio por defecto /etc/procmailrcs de todo el sistema, así como las variables y reglas de entorno. Luego busca si hay un archivo .procmailrc en el directorio principal del usuario para encontrar las reglas específicas de dicho usuario. Muchos usuarios también crean archivos rc adicionales propios para Procmail que utilizan el archivo.procmailrc, pero que se pueden activar o desactivar rápidamente si se produce un problema al filtrar el correo.

Por defecto, no hay archivos rc aplicables a todo el sistema en el directorio /etc y tampoco hay archivos .procmailrc. Para empezar a usar Procmail, deberá crear un archivo .procmailrc con variables y reglas de entorno concretas que expliquen lo que deberá realizarse con determinados mensajes.

En la mayoría de las configuraciones, la decisión sobre si Procmail se arranca e intenta filtrar el correo se basa en el existencia de un archivo de usuario .procmailrc. Para desactivar Procmail, pero guardar el trabajo en el archivo .procmailrc, copie la información a un archivo de nombre similar que utilice el comando mv ~/.procmailrc ~/.procmailrcSAVE. Cuando esté preparado para realizar nuevas pruebas con Procmail, cambie nuevamente el nombre del archivo a .procmailrc. Procmail empezará a funcionar de nuevo inmediatamente.

LISTSERV

Fué escrito originalmente para el sistema operativo VMS de IBM, pero ya ha sido porteada a GNU/Linux y sistemas ÓNIX. En sus inicios sólo se ejecutaba bajo BITNET, pero ya hoy día también corre perfectamente bien en Internet. Es propietariedad de L-Soft. Versiones gratuitas están disponibles pero es un producto comercial y de alto costo.

Ejercicio 9-1: Correo/Email

Este ejercicio le ayudará a practicar los conceptos aprendidos de correo en una red TCP/IP. efectuarse un correo totalmente manual. Intentaremos enviar un correo entrando comandos SMTP directamente en el servidor SMTP. No se proveen soluciones a este ejercicio.

Para poder hacer esta practica contactaremos un servidor de correo conocido por el puerto 25 y usaremos comandos SMTP.

Ejecute un telnet y conectese al puerto 25 del servidor de correo. Deberá ver un saludo estandar de SMTP. Escriba comandos SMTP como se muestra en el ejemplo que sigue, substituya su dirección y el mensaje apropiado:

En este ejemplo usted a conducido una sesión con un SMTP. Afortunadamente que todo este es efectuado por un programa MUA. Pero siempre es bueno hacer este tipo de practica.

Correo anónimo vía telnet

Una manera muy sencilla de enviar correos anónimos sin necesidad de utilizar repetidores es conectarse a un servidor de SMTP a través de Internet, simplemente haciendo un TELNET al puerto 25.

Los pasos a seguir son los siguientes:

1) Localizar un servidor de SMTP lo suficientemente antiguo como para no incluir en las cabeceras del

correo que envía la dirección de la máquina que se le conectó. En el ejemplo, llamaremos a esta máquina aaa.bbb.ccc

Sin duda, éste es el paso más difícil de todos. Sin un servidor así no es posible enviar correos anónimamente. Puedes comprobar los siguientes , a ver si alguno sirve todavía.

2) Busca para ti un nombre cualquiera de máquina (dirección IP), pero que exista. Puede servir uno cualquiera de FTP, HTTP, o lo que sea. Será tu dirección IP falsa, a la que llamaremos xx.yy.zz.

3) Establece una conexión TELNET al puerto 25 con la máquina encontrada en el paso 1.

En GNU/Linux o ÓNIX, escriba:

telnet aaa.bbb.ccc 25

4) Si el sitio acepta la petición de conexión, te aparecerá un mensaje como

220 aaa.bbb.ccc ESMTP Sendmail 8.7.6/8.7.3; Tue, 3 Feb 1998 16:45:30+0100

5) Después de la bienvenida de la máquina, salúdale tú escribiendo:

HELO xx.yy.zz

a lo que el host responderá con alguna clase de presentación, como por ejemplo:

250 aaa.bbb.ccc Hello xx.yy.zz [###.###.###.###], pleased to meet you

6) Escribe los siguientes comandos, sin olvidar el retorno de carro al final de cada línea:

```
MAIL FROM: <unadireccion@falsa.es>
RCPT TO: <tudireccion@verdadera.es>
DATA
Subject: El tema del correo
A continuación el texto del mensaje. No olvides dar un
retorno de carro adicional  después del subject. Todos
los mensajes deben terminar con un punto en una línea sola.
.QUIT
```

Con esto ya habrías enviado un correo y cerrado la sesión con el host. Espera a que te llega el mail y examina las cabeceras para ver si ha quedado rastro de tu dirección de máquina o algo que te delate. Si fuera así, vuelve al paso 1 y busca un servidor de SMTP apropiado.

A modo de ejemplo, puedes utilizar el applet de demostración en .

Utilizando el applet, recibirás algo como

Received: from localhost (tu verdadera máquina) by aaa.bbb.ccc (8.7.6/8.7.3) with SMTP id RAA08608 for ; Tue, 3 Feb 1998 17:02:43 +0100

que desvela tu dirección, por lo ese servidor no puede ser usado para enviar correos anónimos. Sin embargo hay otros que sí que lo son. Sólo tienes que buscarlos.

RESUMEN

En este capítulo, cubrimos los variados protocolos de e-mail y conceptos básicos de correo como es direccionamiento, enrutar y los componentes de una red de correo.

Los tópicos claven incluyen:

- El proceso de correo en sistemas GNU/Linux está dividido en varas piezas:

- El Mail User Agent (MUA), donde el usuario lee y crea los mensajes
- El Mail Transfer Agent (MTA), transfiere mensajes de un sistema a otro.
- El Delivery Agent, cual transfiere los mensajes desde el MTA al MUA.

- DNS asisten en el proceso de enrutamiento de correo proveyendo los records MX
- SMTP (Simple Mail Transfer Protocol) es el protocolo estandar usado para transferir los mensajes.
- POP (Post Office Protocol) usado por un MUA para accesar los mensajes almacenados en el server.
- IMAP (Internet Message Access Protocol) muy parecido al POP, pero tiene más características agregadas como son la posibilidad de crear folders y administrar los mensajes desde el servidor.

PREGUNTAS POST-EXAMEN

Respuestas a estas preguntas se encuentran en el appendice A.

1.- Si el mensaje enviado es retornado, ¿qué pasos deben ser tomados para determinar el problema?

2.- ¿ Cúal es el propósito de los alias?

3.- ¿ Qué puede un servidor de correo hacer para protegerse de spam?



EL NFS (NETWORK FILE SYSTEM)

Objetivos

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Básicamente operar y configurar los daemons de NFS.
- Implementar la compartición de archivos vía NFS.
- Describir la estructura y la aplicación de NFS

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son los beneficios de usar NFS?
2. ¿Cómo es que un cliente de NFS accesa los directorios de otros equipos?
3. ¿Por qué debe usted 'hard mount' el NFS?
4. ¿Por qué es el comando mount único al sistema operativo?
5. ¿Cuándo debe usted montar su NFS?

INTRODUCCIÓN

Sistemas GNU/Linux tiene multiple tipos de sistemas de archivos disponibles. Uno de esos es el NFS (Sistema de Archivos de Red), cual fué desarrollado por Sun Microsystems en el año 1984. El NFS es usado para permitir que multiple computadoras accesen otras estructuras de archivos que se encuentren interconectadas vía TCP/IP. NFS le permite a los usuarios ver estructuras de archivos remotas en una manera transparente. La estructura de archivos remota le aparece al usuario dentro de su estructura propia e igual estructuralmente. NFS le permite a los usuarios accesar más de un sistema de archivos en diferente equipos sin importar que sistema operativo se encuentran ejecutando y bajo que sistema de archivos operan. NFS permite que sistemas de archivos completos y parciales de otros hosts sean montados localmente.

NFS esta compuesto de tres capas para facilitar su uso genérico a través de multiple plataformas. Estas capas son la capa PRC, la cual pasa la data entre los hosts y la capa XDR (la cual provee independencia de la data), y la capa superior, la cual consiste del protocolo Mount y el NFS. Desafortunadamente , NFS no proveé ningún nivel de seguridad, esto debe ser proveído por el sistema operativo.

Uno de los problemas del NFS es que los clientes y servidores se confián incondicionalmente entre sí. Esto puede causar un problema, por ejemplo, la cuenta root ha sido en el pasado en un host cliente. Otro problema es que el nombre de host puede ser engañado (un host hacerse pasar por otro). Para aliviar este problema, podemos usar las opciones nosuid, y root_squash cuando exportamos recursos. Estas opciones requieren que las peticiones procedan desde puertos privilegiados, no permitir ejecución tipo SUID, y mapear acceso de root a anónimo UID, respectivamente. También otra precaución es la de montar lo menos posible sistemas de archivos con las opciones rw. Todos los hosts en el archivo exports deben ser de nombre FQDN (Nombre de Dominios Plenamente Calificados). Esto reducirá el riesgo de enmascararse como otro usuario (spoofing).

NFS es similar a SMB (Server Message Block), cual es usado por Microsoft para compartir archivos, impresoras, puertos seriales y abstracciones de comunicación como son tuberías de nombres y posiciones de correo. NFS, utiliza un protocolo diferente al SMB y sólo comparte archivos. NFS fué desarrollado para sistemas UNiX, aunque ya existen clientes para muchas otras plataformas.

NFS usa una relación cliente/servidor para manejar la compartición de archivos. Cualquier máquina puede ser el servidor y el cliente a la misma vez. Pero ningún equipo puede ser el cliente y el servidor para la misma estructura de archivo.

LO BÁSICO DEL NFS

En esta sección discutiremos los elementos básicos del NFS. Estudiaremos los principios del NFS, que se requiere tanto de un cliente como de un servidor, que protocolos son usados y cuales son las limitaciones del NFS.

El NFS nos presenta como si un sistema de archivos en un host remoto estuviese realmente como si fuese parte de sistema de archivos local. En este aspecto, nos proveé las funcionalidad generalmente asociadas con los sistemas operativos de redes (NOS), así como es NetWare de Novell.

NFS nos permite la centralización de la administración de los discos. Donde en vez de duplicar directorios comunes de trabajo en cada sistema, NFS nos proveé una copia única de los directorios que son compartidos por todos los sistemas en la red. Esto significa que sólo habrá que efectuar un sólo y único backup, ya que los clientes no almacenan data localmente. Aún yendose más lejos, si las aplicaciones también están almacenadas en el servidor, la instalación y actualización de estas será limitado del lado del servidor.

Como hemos de ver más adelante en este mismo capítulo, una buena característica del NFS, es su habilidad de poder recuperarse de fallas del sistema sin conocimiento de las aplicaciones del nivel del sistema, claro esta sin tomar en cuenta el tiempo de reiniciar el servidor.

En esta sección discutiremos los siguientes temas:
- Principios
- Terminología
- Servidor NFS
- Clientes NFS

Principios

El NFS fué originalmente desarrollado como una adición al sistema operativo de Sun MicroSystems Solaris, cual es un derivado de UNiX. La idea inicial fué que los archivos tanto en el sistema local como el remoto, aparentarán al usuario idénticos. Para mejor entender los comandos asociados con el NFS, es muy apropiado primero estudiar como funcionan los sistemas de archivos UNiX en su totalidad.

Al instalar un sistema GNU/Linux, el administrador del sistema dividirá el espacio disponible en secciones de discos o particiones. Esto facilita la administración del espacio en disco para los propósitos de backup, etc. Si existe sólo un disco, las particiones residirán todas en este disco; si son más de uno estas se repartirán entre los discos. Cada partición tendrá una estructura impuesta a ella que soporte la creación y eliminación de archivos y directorios de GNU/Linux. Esta estructura se denomina un sistema de archivos GNU/Linux.

Aún con particionado o distribución física de los discos, el árbol del Sistema GNU/Linux le aparecerá al usuario como un sólo y único conjunto de archivos y directorios. Para lograr este efecto, cada uno de estos sistemas de archivos es amarrado o montado, en una parte del árbol de archivos al inicio del sistema. El amarre de uno de sistemas de archivos se efectúa con el comando mount. Los argumentos para este comando son las diferentes particiones que desea montar y la parte de donde en el sistema de archivos de donde usted desea montarlo. Una vez montado un sistema de archivos, otros sistemas de archivos pueden ser montados dentro y por debajo de este.

Al inicio del sistema, el sistema de archivos GNU/Linux consiste solamente de un punto de montaje en la cima del árbol. Todos los archivos y directorios que normalmente vemos se encuentran en esa posición porque los scripts de inicio de GNU/Linux efectúan el comando mount para construir este árbol cada vez que se inicia el sistema.

La idea detrás del NFS es que los directorios que se exporten aparezcan como parte del árbol del sistema de archivos GNU/Linux. Por esta razón es que el comando mount es retenido para manipular el NFS. La única modificación es que en vez de una partición física en el equipo local, el administrador específica el nombre de un host en el que los archivos están y donde en ese host los archivos residen. El conjunto de archivos en el sistema remoto se denominan como la fuente (source). El administrador también deberá especificar el tipo de sistema de archivos que se va a montar. Al usar NFS, este siempre será nfs.

Una vez montado, los archivos remotos aparecerán iguales que los archivos locales al usuario. Los mismos principios aplican a sistemas que no sean GNU/Linux, la única diferencia siendo que los archivos importados aparecerán en un estilo apropiado al sistema, Ej., bajo DOS, OS/2 y NT aparecerán como un nuevo dispositivo (drive) lógico. Así que puede verlo similar al proceso de mapeo remoto en los sistemas operativos de redes para PC.

Terminología

Un sistema que contiene archivos y directorios que van a ser compartidos con otros sistemas es conocido como un servidor de NFS. El servidor debe marcar los archivos y directorios que van a ser compartidos. Estos son conocidos como los recursos (resources) compartidos por el servidor NFS. El proceso de marcar los recursos que van a ser compartidos es conocido como exportar o compartiendo los recursos.

Un sistema que importa o monta recursos desde un servidor NFS es conocido como un cliente. Un cliente NFS sólo puede importar un recurso si este ha sido exportado por servidor NFS. así que el servidor mantiene el control sobre que parte del sistema de archivo(s) es visible a los clientes.

Es posible que un único sistema actúe como ambos el servidor y el cliente NFS, aunque re-exportar recursos que han sido importados no es permitido.

Servidor NFS

Para ser un servidor NFS, el administrador del host debe primero identificar que parte de disco o recursos el desea compartir. Estos recursos deben entonces ser exportados por el servidor. Esto permite que clientes identifiquen cuales recursos remotos ellos están permitidos a importar vía NFS.

Lo que sigue son las restricciones de que puede ser exportado por servidores NFS:

- Cualquier sistema de archivos o parte de un sistema de archivos, puede ser exportado. Un subconjunto apropiado de un sistema de archivos es un subdirectorio en la jerarquía de archivos que se encuentra debajo del punto de montaje de ese sistema de archivos. Por ejemplo, si /home es un sistema de archivos que contiene un subdirectorio /home/trabajos entonces este es considerado un subconjunto apropiado.

  Esto permite que parte de un sistema de archivos más grande sea selectivamente exportado sin tener que poner disponible todo el sistema de archivos.

- Si un directorio es exportado, entonces un subdirectorio sólo podrá ser exportado si este existe en un dispositivo diferente. Esto es para prevenir posibles confusiones si un intento es hecho de re-exportar archivos y directorios que ya han sido exportados.

- Si un directorio es exportado, entonces un directorio padre sólo podrá ser exportado si existe en un dispositivo diferente. Esta es contraria de la restricción anterior y es para prevenir que un sistema de archivo completo sea exportado cuando ha sido previamente restringido al directorio nombrado.

- Solamente recursos locales pueden ser reportados. Es posible re-exportar un recurso que ha sido importado desde otro sistema de archivos.

Cliente NFS

El host cliente primero debe identificar al host servidor que exporta los recursos que el cliente desea utilizar. Si el servidor ofrece recursos exportados, el cliente puede elegir cuales recursos este desea importar. Entonces ejecutaría el comando de importar estos recursos en una parte de su sistema de archivos.

Una vez importado, el recurso remoto aparecerá al usuario igual que los archivos y directorios. Como aparece el recurso importado dependerá del sistema al cual se importa.

No se impone restricción en el número de recursos que un cliente puede importar desde un servidor o varios servidores. Además, un host puede actuar como ambos tanto el cliente como el servidor. En un caso importando y en el otro exportando recursos.

Hay varias opciones que pueden ser establecidas al importar un recurso remoto. Por ejemplo, un cliente

puede importar un recurso en modo de solo lectura (read-only) aunque el servidor las haya exportado de lectura y escritura (read/write).

El cliente también puede especificar el tipo requerido a importar. Hay dos puntos en esto, hard y soft. La opción del montado hard es recomendada para los recursos que se montan en modo de lectura y escritura (como son los directorios home). Con un hard mount, el cliente continuará a reintentar la operación hasta que el servidor responda. Si un proceso esta escribiendo a un archivo y el servidor falla, la escritura del archivo continuará cuando el servidor regrese, y no habrá perdida de data. Un soft mount causará que el cliente se rinda si el servidor no responde después de un tiempo dado. En este caso, el recurso se removerá del árbol o la lista de dispositivos lógicos.

En sistemas multiusuarios, la habilidad de importar recursos esta restringidas a los administradores del sistema. Los recursos remotos por lo general se montan al momento de arranque.

PROTOCOLOS NFS

El NFS consiste de tres capas distintas y separadas. Dos de estas son los bloques son usados por los protocolos de alto nivel. Ellos proveen una interfaz de RPC (Remote Procedure Calls) para pasar data entre los hosts y una capa independiente de máquina llamado XDR (External Data Representation).

Las capas superiores de NFS consisten de dos diferentes protocolos. El primero es el protocolo Mount, este es usado para negociar la importación del recurso remoto entre el cliente y el servidor. Ya que la estructura de de los diferentes sistemas de archivos difiere de un sistema operativo a otro este protocolo varia de un sistema operativo a otro. El otro protocolo es el NFS, este es independiente del sistema. Esta tiene que ver con el manejo de pasar los bloques de data entre los clientes y el servidor.

NFS se ejecuta sobre UDP. Esto le puede parecer extraño debido a la no confiabilidad ligada a UDP. Pero, si le ofrece a NFS una combinación útil de atributos. El protocolo NFS es:

- **Stateless (Sin Estado)**
  Esto se refiere a que el servidor NFS no mantiene una noción de cuales clientes están en la actualidad usandolo, o cuales archivos están siendo accesados.
- **Idempotent (Idempotente)**
  La mayoría de las peticiones pueden ser hechas más de una vez al servidor, sin afectar al servidor.

En combinación estos atributos le dan a NFS una fuerte resistencia a la perdida de data debido a las fallas de de los servidores. Como no existe un estado, el cliente no detecta la ausencia del servidor y vice versa. También, debido a su idempotencia, el cliente puede mantenerse intentando la petición hasta que el servidor se recupere. El cliente solo ve que el servidor es lento.

Los siguientes tópicos serán discutidos en esta sección:

- Pila/Stack de Protocolos
- RPCs (Remote Procedure Calls)
- XDR (External Data Representation)
- Protocolos NFS de Nivel Alto
- Limitaciones del NFS

Pila/Stack de Protocolos

NFS está formado por 4 protocolos distintos. Cada uno depende de las RPC y de portmap (también llamado rpc.portmap). Un portmapper convierte números de programa RPC en números de puerto. Cuando un servidor RPC se inicia, dice a portmap qué puerto usará y el número de programa RPC manejado. Cuando un cliente quiere enviar una petición RPC a un número de programa dado, primero contacta con el servidor portmap para tomar el número de puerto dando acceso al programa deseado. Después, dirige los paquetes RPC al puerto correspondiente.

Los 4 servicios que permiten funcionar a NFS son:

Protocolo	Descripción	Demonio
nfs	Este protocolo es el básico y permite crear, buscar, leer o escribir archivos. Este protocolo también maneja autentificación y estadísticas de archivos.	**nfsd**
mountd	Éste se encarga de montar sistemas exportados para acceder a ellos con nfs. El servidor recibe peticiones como mount y umount debiendo mantener información sobre los sistemas de archivos exportados.	**mountd**
nsm (Network Status Monitor)	Se usa para monitorizar los nodos de la red y así conocer el estado de una máquina (cliente o servidor). Informa, por ejemplo, de un rearranque.	**statd**
nlm (Network Lock Manager)	Para impedir modificaciones de los datos por varios clientes al mismo tiempo, este protocolo maneja un sistema de bloqueo. Así, con la ayuda del protocolo Nsm es posible conocer cuándo se está reiniciando un cliente. Nsm libera todos los bloqueos del cliente antes de devolverlos.	**lockd**

El demonio knfsd, disponible con las últimas versiónes del núcleo, soporta directamente los protocolos nfs y nlm. Por otro lado, mountd y nsm no están todavía soportados. Cuando el servidor NFS está instalado y arrancado, podemos verificar que todo esté funcionando con el comando:

```
>> ps auxwww | egrep "nfs|mount|lock|stat"
root    1370  0.0  0.2  1176  580 ?      S    22:28  0:00 rpc.mountd --no-nfs-version 3
root    1379  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1380  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1381  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1382  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1383  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1384  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1385  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1386  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [nfsd]
root    1399  0.0  0.0     0    0 pts/0  SW   22:28  0:00 [lockd]
root    1409  0.0  0.2  1156  560 ?      S    22:28  0:00 rpc.statd
root    1652  0.0  0.1  1228  484 pts/3  S    22:49  0:00 egrep nfs|mount|lock|stat
```

RPCs (Remote Procedure Calls)

Los RPCs son análogos a las llamadas de procedimientos local (LPC). Con una llamada local, el que llama el procedimiento, lo hace pasandole parámetros; el procedimiento ejecuta cierta acción y el procedimiento retorna al que llama alguna información. Los RPCs permiten que la acción sea efectuada en un equipo remoto por un servidor RPC.

El cliente hace llamadas de procedimiento locales, las cuales hacen peticiones al servidor. Cuando estas peticiones arriban, el servidor hace un llamado a una ruta que despacha, efectúa cualquier servicio que ha sido requerido, envía una repuesta y la llamada de procedimiento es retornada al cliente.

La aplicación cliente, simplemente ha llamado un procedimiento local que contiene detalles oculto del mecanismo subliminar de la red. No es necesario que el cliente ni siquiera este consiente de la existencia de la red.

El NFS usa el mecanismo ONC (OPEN Network Computing) de Sun Microsystems para efectuar RPCs. Esto está definido en el RFC 1057, el cual no es un estándar de Internet, sino para el beneficio de las personas que desean dar uso de este protocolo.

Más adelante observaremos que los servidores RPC no usan el método tradicional TCP/IP de escuchar sobre puertos bien conocidos. Sino que ellos usan un tercero llamado port mapper, para facilitar la comunicación entre el servidor y el cliente.

Procedimientos remotos son identificados por un número de programa y de procedimiento. El número de programa específica un grupo de procedimientos relacionados, mientras que el número de procedimiento identifica un procedimiento dentro del grupo. Cada conjunto de procedimiento tiene un número de versión. Cada característica nueva del servidor sólo necesita tener nuevas funciones escritas; un número de programa nuevo no tiene que ser asignado.

Un cliente hace una llamada a un procedimiento remoto especificando su número de programa, número de versión y número de procedimiento.

La mayoría de servicios, como son telnet, FTP y el correo, son dueños de números de puertos bien conocidos, los cuales sus clientes usan para intercomunicarse con ellos. Este es un puerto de la capa de Transporte, como es un puerto TCP o UDP.

Un servidor RPC normalmente no elige un puerto específico de la capa de Transporte por varias razones:
- RPC esta diseñado para ser independiente de la capa de Transporte (lo que significa que al RPC no le importa como un mensaje se pasa de un proceso a otro); amarrar una dirección de transporte restringiría esa libertad.
- Los puertos no pueden ser colocados dinámicamente.
- Los números de Puertos son pocos relativamente.

Los números de puertos son asignados dinámicamente por un servidor y un mecanismo es establecido para los clientes que descifra exactamente cual puerto es.

Los clientes descubren el puerto del servidor así:

1.- Al iniciar un servidor RPC, este peticiona un número de puerto al equipo host. Este puede ser cualquier número de puerto; a; servidor no le importa que cual. El servidor luego le informa al port mapper de la versión, protocolo de transporte y el número de puerto en uso por el programa. El port mapper almacena este record en un mapa desde el (programa, versión, protocolo) al (número de puerto). El port mapper mantiene esta información de cada servidor RPC en ese host.

2.- El programa cliente ubica el programa, la versión y el protocolo localmente y se lo envía en forma de una petición al port mapper. El port mapper responde con el número de puerto.

3.- El cliente ahora se comunica con el servidor RPC directamente.

En UNiX, el port mapper es implementado o con el servidor rpcbind o portmap.

El comando nfsstat nos puede dar estadísticas de las llamadas RPC del cliente y el servidor y los proce-

sos NFS del cliente y servidor. Por defecto, nfsstat nos devuelve toda la información. Con la opción -u hace una llamada UDP, y con la opción -t efectúa una llamada TCP al probar procedimientos RPC null.

XDR (External Data Representation)

El XDR se ocupa del problema de enviar tipos de data compleja entre equipos de diferente arquitecturas. Claro esta, con tipos de data simple como el texto ASCII, no existen problemas. El problema ocurre al enviar tipos complejos como los enteros largos, estructuras o records, arreglos o estructuras de datos arbitrarías construidas en base a este tipo de datos. Tipos de datos recibidos en una máquina con un tipo de arquitectura diferente de procesador de ese que envía el mensaje, corren el peligro de ser mal interpretadas debido al ordenamiento de los bytes. Aún más lejo es que el alineamiento de los limites de los words puede causar que el tamaño de una estructura varía de una máquina a otra; los punteros no tienen significado fuera del equipo que fueron definidos.

El XDR resuelve los problemas de portabilidad a través de la estandarización de la data que se envía por la red. Este define un orden único, una única representación de punto flotante, y así sucesivamente.

Los programas traducen data al formato canónico del XDR y la envían a través de la red; el receptor traduce la data a su representación local. El XDR permite estructuras de datos arbitrariamente complejas (hasta punteros a otras estructuras) que sean transmitidas con portabilidad absolutas.

Protocolos NFS de Nivel Alto

En la capa superior del NFS hay dos protocolos por separado. El primero de estos dos son el Mount, el cual es responsable de negociar el el mapeo del volumen lógico remoto en los términos que el cliente lo pueda entender. Por esta razón el protocolo Mount es dependiente del sistema operativo y existe para soportar la importación de recursos remotos. Esto implica que los archivos son vistos diferente de un sistema operativo a otro. Por esto es que el server pueda que ejecute un servidor por separado para manejar los diferentes protocolos, Ej., pcnfsd.

El protocolo NFS es totalmente independiente del sistema operativo. Esto es relativamente fácil ya que el protocolo solo se concierne con la transferencia de archivos de data de un sistema a otro. La versión anterior, la versión 2, definida en el RFC 1094. La versión 3 de hoy día esta definida en el RFC 1813. Ambos documentos son pura documentación del producto más que un esfuerzo de estandarización de tecnología.

Limitaciones del NFS

La apertura y la característica sin estado (stateless) del NFS dificultan su mantenimiento en algunos sistemas operativos, en particular el caso de GNU/Linux. Un ejemplo de un efecto secundario de poder usar NFS con otros sistemas operativos como es el VMS, desde GNU/Linux significa que los vínculos simbólicos de GNU/Linux no serán soportados en el otro sistema operativo, al menos que el otro SO los soporte. Otro ejemplo es que se puede eliminar archivos mientras se encuentran en uso. Un programa de GNU/Linux puede hacerlo para crear un archivo temporal; el archivo se abre, luego se elimina. Este aún es accesible por el programa, pero ya no tiene un nombre en el sistema de archivos; GNU/Linux no libera el bloque en el disco hasta que el archivo este finalmente cerrado. Brindar soporte a todo esto por el protocolo NFS significaría introducir estado en el servidor; esto si puede ser soportado en el cliente.

Acceso a los archivos de dispositivos, como son los archivos especiales de GNU/Linux y de dispositivos DOS de LPT, no es soportado sobre el NFS. Cerrar acceso a los archivos (File Locking) es difícil en un ambiente de red; ya que estuviéramos de nuevo tratando de introducir estado en un sistema diseñado para ser sin estado. Existe un demonio por separado (Lock-Manager Daemon) que administra el cerrado de los archivos.

Algunas de las decisiones de diseño limitan el conjunto de aplicaciones para el cual NFS es apropiado:

- El modelo de obtención (caching) asume que la mayoría de archivos no serán compartidos. La funcionalidad sufre cuando los archivos son compartidos pesadamente.
- El protocolo sin estado requiere algunas perdidas de semánticas de UNiX tradicional. El cerrar sistemas de archivos (flocks) ha de ser implementado por un demonio de estado separado. Aplaza la liberación de espacio en un archivo diferenciado hasta que el proceso final ha cerrado el archivo es aproximado con una heurística que a veces falla.

A pesar de estas limitaciones, NFS ha proliferado porque este hace un razonable trato entre semántica y funcionalidad; su bajo costo de adopción lo ha hecho ubicuo.

CONFIGURAR EL NFS

Aquí examinaremos los pasos básicos necesarios para poner en ejecución un servidor y un cliente NFS en varios ambiente. Los siguientes tópicos son discutidos en esta sección:

- Exports/Exportar
- Mounting/Montar

Exports/Exportar

Exportar es simplemente la acción de hacer disponible un directorio o archivo local para que pueda ser accesado a través de la red por los clientes ejecutando NFS. Estos directorios o conjunto de archivos son conocidos como los recursos compartidos.

El servidor NFS mantiene una tabla de los recursos exportados. Dependiendo en la implementación la lista o tabla pueda que resida o en el kernel o en el daemon de mount.

La tabla es inicializada al leer un archivo de texto que contiene la información a cerca de los recursos a exportar. Si cualquier detalle de recursos exportados cambia, este archivo tendrá que ser modificado concordantemente, y la lista debe ser restablecida. El superusuario puede utilizar comandos interactivos para agregar o eliminar recursos desde la lista de exportación.

En GNU/Linux, como en todos los UNiX, el archivo que contiene la lista de los recursos a exportar se encuentra en /etc/exports. Este es un archivo de texto sencillo parecido al /etc/fstab. NFS ofrece varios comando para su manipulación desde la línea de comandos. Entre estos se encuentran los siguientes:

```
$ exportfs        # nos muestra la tabla de los recursos actual exportada
$ exportfs -a     # Exporta todos los recursos en el archivo /etc/exports
```

El Archivo /etc/exports y sus Opciones

El archivo exports es un archivo de texto simple que específica esos recursos que pueden ser exportados, así como ciertas características adicionales de estos recursos. Este archivo controla cuáles sistemas de archivos son exportados a las máquinas remotas y específica opciones particulares que controlen todo. Las líneas en blanco son ignoradas, se pueden comentar líneas con el símbolo # y las líneas largas pueden ser divididas con una barra invertida (\). Cada sistema de archivos exportado debe tener su propia línea. La lista de máquinas autorizadas colocada después de un sistema de archivos exportado, debe estar separada por un espacio. Las opciones para cada uno de las máquinas deben ser colocadas entre paréntesis directamente detrás del identificador de la máquina, sin ningún espacio de separación entre la máquina y el primer paréntesis. Cada host puede ser seguido de una lista de opciones encerradas dentro de paréntesis, como estas:

ro

Sólo lectura (read-only). Las máquinas que monten este sistema de archivos no podrán cambiarlo. Para permitirlas que puedan hacer cambios en el sistema de archivos, debe especificar la opción rw (lectura-escritura, read-write).

async

Permite al servidor escribir los datos en el disco cuando lo crea conveniente. Mientras que esto no tiene importancia en un sistema de sólo lectura, si una máquina hace cambios en un sistema de archivos de lectura-escritura y el servidor se cae o se apaga, se pueden perder datos. Especificando la opción sync, todas las escrituras en el disco deben hacerse antes de devolver el control al cliente. Esto puede que disminuya el rendimiento.

wdelay

Provoca que el servidor NFS retrase el escribir a disco si sospecha que otra petición de escritura es inminente. Esto puede mejorar el rendimiento reduciendo las veces que se debe acceder al disco por comandos de escritura separados. Use no_wdelay para desactivar esta opción, la cual sólo funciona si está usando la opción sync.

root_squash

Previene a los usuarios root conectados remotamente de tener privilegios como root asignándole el userid de 'nobody'. Esto reconvierte el poder del usuario root remoto al de usuario local más bajo, previniendo que los usuarios root remotos puedan convertirse en usuarios root en el sistema local. Alternativamente, la opción no_root_squash lo desactiva. Para reconvertir a todos los usuarios, incluyendo a root, use la opción all_squash. Para especificar los ID de usuario y grupo para usar con usuarios remotos desde una máquina particular, use las opciones anonuid y anongid, respectivamente. De esta manera, puede crear una cuenta de usuario especial para usuarios NFS remotos para compartir y especificar (anonuid=<uid-value>,anongid=<gid-value>), donde <uid-value> es el número ID de usuario y <gid-value> es el número ID de grupo.

Note que la revisión de un recurso estar de sólo lectura no es hecha en el momento de mount pero cuando una operación requiere permisos de escritura. Esto es porque es posible para un cliente montar un recurso en solo lectura (ro) aunque se ha exportado lectura y escritura (rw). Así que no siempre es posible determinar con precisión que operaciones son permitidas.

Aquí se le presenta un ejemplo del archivo /etc/exports. Este exporta los directorios home de todas las computadoras y un sistema root completo para los clientes diskless, de nombre cliente1, cliente2 y cliente3.

```
/exportar/diskless-root     diskless*.abiertos.org    (rw,no_root_squash)
/home                       *.abiertos.org            (rw)
/home/musica                192.168.0.2               (ro)
/home/exportar              192.168.0.2
```

Mounting/Montar

El comando mount opera diferente al montar volumenes NFS que cuando son de sistemas de archivos locales. El sintaxis de comando es así:

```
$ mount -t nfs -o listaopciones nombre_host:/exportar /ruta_local
```

En vez de el nombre de un dispositivo, se da un par de hostname:/exportar. El nombre de host debe referirse a un sistema que puede ser contactado por el sistema local. La referencia /exportar debe ser un recurso en el sistema remoto. Podemos especificar opciones en un lista separada por comas.

Opciones de Mount para las Entradas en /etc/fstab

La opción de hard mount es recomendada para los sistemas de archivos que son montados lectura/escritura (como los directorios home de los usuarios). Con el hard mount, el cliente continuará reintentando las operaciones hasta que el server le responda. Si un proceso esta escribiendo a un archivo y el servidor falla, la escritura continuará cuando el servidor retorna, y no se perderá ninguna de la data. Con el soft mount, el cliente se rendirá si el servidor no responde después de cierto tiempo, y la escritura entonces nunca se llevará acabo.

La opción intr asegura que los procesos accediendo los hard mounts pueden ser interrumpidos si el servidor se cae y no se espera que retorne inmediatamente. Esto es casi siempre lo más apropiado a hacer.

Si un sistema de archivo es montado regularmente, particularmente si debe ser montado al momento de boot, debe colocar una entrada en el /etc/fstab. Al momento de iniciar el NFS en el cliente, este tratará montar el recurso especificado. El comando mount -a -t nfs (debe colocarlo en rc.local) se asegurará que todos los recursos en el /etc/fstab sean montados. Usted debe notar que los por defecto de los montados de NFS no son optimo y puede llevarnos a problemas de rendimiento debido a los falla por reintento. Una entrada correcta en el fstab para automontar un volumen NFS es parecido a este:

hostname:/dir_exportado nfs rsize=8192,timeout=14,intr 0 0

Fijese que el tamaño de bloque de lectura y escritura por defecto es de 1024, cual es muy pesimista. Pero, valores muy grandes de rsize puede causar problemas de perdida de paquetes y su transmisión subsequente. El valor de 8192 funciona bien en las mayoría de situaciones.

Colocando una línea adecuadamente formada en el archivo /etc/fstab tiene el mismo efecto que el montaje manual del sistema de archivos exportado. El archivo /etc/fstab es leído por el script /etc/rc.d/init.d/netfs cuando arranca el sistema y cualquier compartición NFS listada será montada.

Un ejemplo de línea /etc/fstab para montar un NFS exportado será parecida a:

<server> : </path/of/dir> </local/mnt/point> nfs <options> 0 0

La opción **<server-host>** tiene que ver con el nombre de la máquina, dirección IP o nombre de dominio totalmente cualificado del servidor que exporta el sistema de archivos.

La opción **</path/of/directory>** es la ruta al directorio exportado.

La opción **</local/mount/point>** específica dónde montar en el sistema de archivos local el directorio exportado. Este punto de montaje debe existir antes de que /etc/fstab sea leído o el montaje fallará.

La opción nfs específica el tipo de sistema de archivos que esta siendo montado.

El área **<options>** específica como el sistema de archivos es montado. Por ejemplo, si las opciones indican rw,suid, el sistema de archivos exportado será montado en modo de lectura-escritura y los ID de usuario y grupo puestos por el servidor serán usados. Aquí no se usan paréntesis.

Existen otras numerosas opciones para afinar montados NFS. Deberá consultar las páginas man de los comandos y archivos especiales nfs, mount, y el fstab.

Ejercicio 10-1: Crear un Recurso Exportado NFS

En este ejercicio exportaremos el directorio home de una computadora y la montaremos en otra. La máquina que compartirá sus archivos la llamaremos el server, mientras que la máquina que monta el recurso llamaremos el cliente. Será necesario acceso a la cuenta de root para llevar acabo este ejercicio. No se proveen soluciones para este ejercicio.

1.- Especifiquemos que recurso queremos compartir con el cliente. Elegiremos el directorio del usuario completo /home/usuario. Debemos editar el archivo /etc/exports en el servidor y le agregaremos la siguiente línea:

/home/nombre_usuario 192.168.2.7 (rw)

Esta sentencia compartirá el directorio /home/nombre_usuario con el computador cuyo <IP> es 192.168.2.7. Una vez el directorio es montado en el cliente, este tendrá acceso de de lectura y escritura.

2.- Paremos y reiniciemos el servidor NFS:
/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start

3.- Edite el archivo /etc/fstab en el cliente para montar en directorio en el Cliente. Agregue la siguiente línea en el archivo /etc/fstab en la máquina Cliente:
192.168.2.7:/home/nombre_usuario /mnt/dir_nfs nfs defaults 0 0

4.- Crearemos el directo donde vamos a montar en la máquina Cliente el recurso compartido:
mkdir /mnt/dir_nfs

5.- Montar el recurso compartido en el directorio /mnt/dir_nfs en la máquina Cliente:
mount /mnt/dir_nfs

6.- Navegue al directorio /mnt/dir_nfs en la máquina Cliente a ver si monto correctamente:
cd /mnt/dir_nfs

7.- Cree un archivo nuevo dentro del directorio /mnt/dir_nfs en la máquina Cliente para asegurarse que tiene permisos de lectura y escritura:
touch mi_archivo.txt

ASEGURAR NFS

NFS trabaja muy bien compartiendo sistemas de archivos enteros con un gran número de máquinas conocidas de una manera muy transparente. Muchos usuarios que acceden a archivos sobre un punto de montaje NFS pueden no estar atentos a que el sistema de archivos que están usando no está en su sistema local. Sin embargo, esta facilidad de uso trae una variedad de potenciales problemas de seguridad. A pequeña escale NFS no presenta grandes riesgo de seguridad pero al implementarse en gran escala el uso de NFS tanto en los servidores como en los clientes si presenta unos aspectos que deben tomarse en cuenta.

Los puntos siguientes deberían ser considerados cuando se exporte sistemas de archivos NFS en un servidor o cuando se monten en un cliente. Haciendo esto reducirá los riesgos de seguridad NFS y protegerá mejor los datos en el servidor. Los siguientes tópicos serán discutidos en esta sección:
• Seguridad y NFS
• Uso Apropiado de NFS

Seguridad y NFS

Ya no debe ser noticia que NFS es inaceptablemente inseguro en las mejores de circunstancias, lo que significa servidores y clientes bien configurados y sin problemas (bugs) y los ports mappers. NFS usa autentificación simple basada en direcciones IP y ID de usuarios. Es relativamente fácil convencer el servidor NFS que eres alguien quien no es, incluyendo personificando a root. Aunque este habilitada la opción de squashing a root en su sistema de archivo exportado, un usuario puedo hacerse pasar como cualquier usuario, no necesariamente root, sin importar el nivel de privilegio del usuario. Peor aún es que podemos hablar directamente con el servidor NFS, obviando el sistema operativo del host completamente; usted simplemente necesita saber como programar el protocolo NFS o encontrar una herramienta que sabe como hablarle.

Para hacer NFS más seguro debemos trabajar más disciplinadamente. Presentaremos algunas informaciones que harán su uso de NFS más seguro. Primero use un esquema de autentificación seguro y habilite el root squashing (mapeando el UID 0 a “nobody”) y talves squashing a UIDs específico que no sean ciertos de confianza. Ejemplos de esquemas de autentificación segura son Kerberos, el cual NFS conoce nativamente, y de

Sun Microsystems RPC seguro. NFS usa RPC para todas sus acciones de autentificación; Secure RPC agrega la encriptación DES al proceso y mucho menos vulnerable. Lo malo de esto es que por lo general requiere que se instale NIS para poder ser mantenido automáticamente. El Secure RPC puede ser usado en un sistema sin NIS, pero el esfuerzo de administración para su mantenimiento y sincronización de la llave y los archivos ID en todas las máquinas es imposible. Hay que mantener en mente que la encriptación DES puede ser vulnerada con el método de fuerza bruta, aunque esto no es una tarea fácil.

Asegurece que su NFS está configurado para sólo escuchar en los puertos de números bajo, los que por lo general son reservados para los procesos privilegiados. Esto va ha prevenir que una herramienta de cracking se comunique directamente con el servidor NFS. Cuidese que esta practica puede afectar algunos clientes Como-UNiX, así que tendrá que tomar ciertas medidas.

Implementaciones NFS varían en calidad. Si está obligado a usar NFS en su redes UNiX tendrá que tener disciplina en sus asignaciones de permisos.

NFS controla quien puede montar y exportar sistemas de archivos basados en la máquina que lo pide, no el usuario que utilizará el sistema de archivos. Las máquinas tienen que tener los derechos para montar los sistemas de archivos exportados explícitamente. El control de acceso no es posible para usuarios, aparte de los permisos de archivos y directorios. En otras palabras, cuando exporta un sistema de archivos vía NFS, cualquier usuario en cualquier máquina remota conectada al servidor NFS puede acceder a los datos compartidos. Para limitar estos riesgos potenciales, los administradores sólo pueden permitir acceso de sólo-lectura o reducir a los usuarios a un usuario común y groupid. Pero estas soluciones pueden impedir que la compartición NFS sea usada de la forma en que originalmente se pensó.

Adicionalmente, si un atacante gana el control del servidor DSN usado por el sistema que exporta el sistema de archivos NFS, el sistema asociado con un nombre de máquina concreto o nombre de dominio totalmente cualificado, puede ser dirigido a una máquina sin autorización. En este punto, la máquina desautorizada es el sistema que tiene permitido montar la compartición NFS, ya que no hay intercambio de información de nombre de usuario o contraseña para proporcional seguridad adicional al montaje NFS. Los mismos riesgos corre el servidor NIS, si los nombres de red NIS son usados para permitir a ciertas máquinas montar una compartición NFS. Usando direcciones IP en /etc/exports, esta clase de ataques son más difíciles.

Los comodines o metacaracteres deben ser usados lo menos posible cuando garantizamos el acceso a una compartición NFS. El uso de los comodines puede permitir el acceso a sistemas que puede no saber que existen y que no deberían montar el sistema de archivos.

Una vez que el sistema de archivos es montado como lectura-escritura por una máquina remota, la única protección que tiene cada archivo son sus permisos. Si dos usuarios que comparten el mismo valor de userid montan el mismo NFS, ellos podrán modificar sus archivos mutuamente. Adicionalmente, cualquiera con acceso root en el sistema cliente puede usar el comando su - para volverse un usuario que tenga acceso a determinados archivos a través de la compartición NFS.

El comportamiento por defecto cuando se está exportando un sistema de archivos a través NFS es usar root squashing. Esto coloca el userid de cualquiera que esté accesando la compartición NFS como el usuario root en su máquina local al valor de la cuenta de 'nobody'. Nunca desactive el aplastamiento (squashing) de root.

Si se está exportando una compartición NFS como de sólo lectura, considere usar la opción all_squash, la cual hace que todos los usuarios accesando el sistema de archivos exportado tomen el userid del usuario nobody.

Uso Apropiado de NFS

No podemos jamas pensar que por la parte de inseguro que NFS es no lo usaremos. En redes UNiX es casi imposible escaparse sin usar NFS. Al usar NFS debe volver a lo básico: propiedad de los archivos, arquitectura de la red y el uso del sistema.

Pueda ser que tenga terminal servers (unidades sin discos) de UNiX o GNU/Linux o cualquier otro tipo de estaciones de trabajo con X en su red. En esta situación estará forzado a usar NFS. Si usa NFS para compartir archivos en una área segura, o para ejecutar una aplicación desde un servidor en vez de localmente, puede ser que usted investigue soluciones alternativas. Espacio en disco es muy económico hoy en día, y tecnologías de aplicaciones como rdist mejora todos los días.

Pero para esas situaciones donde NFS es una necesidad absoluta en un ambiente seguro, aquí le presentamos algunas ideas a recordar cuando configure el servidor:
- Históricamente NFS ha sido vulnerable
- Asegúrese que su por mapper sea sin errores de vulnerabilidad
- Aprenda el historial de su software NFS en particular
- Aprenda todas las opciones del daemon y que hacen
- Solo exporte lo que sea absolutamente necesario
- Mantengase vigilante de los logs del sistema. Los exploits de NFS es la técnica favorita de los crackers.

DIAGNOSTICAR FALLAS

Nada trabaja siempre al 100%, y NFS no es diferente. Problemas han de surgir como en toda las demás cosas, y debemos estar preparados. Afortunadamente en poca ocasiones sucede perdida de data y muy a menudo son problemas de conexiones físicas o de redes. Muchas veces son problemas transitorios y se corrigen solos.

Por lo general los problemas de NFS son síntomas de trastornos en la red. Antes de involucrarse en profundizar en la búsqueda de los problemas de NFS, asegúrese de que su daemon de port mapper esté bien configurado. El NFS dependen de manera muy crítica del port mapper. Usted también debe entender que servidores de RPC se encuentran ejecutando en su computador ya que ellos pueden interactuar negativamente con el NFS si su comportamiento es errático.

Tenemos disponible varias herramientas para detectar problemas que ocurren con el NFS:

rpcinfo	Nos informa del estado de los servidores RPC.
exports	Exporta, re-exporta o elimina de la lista de exportado sistemas de archivos; nos informa que exportamos actualmente.
nfsstat	Nos rinde información estadística concerniendo RPC y NFS.
showmount	Nos informa de los clientes que en la actualidad tienen montado volumenes NFS en el equipo local.

Claro esta, también se encuentran disponible las herramientas de redes que acostumbramos a usar para asistencia y diagnóstico de redes como son, netstat y el ping. Estas herramientas pueden ser usadas para revisar las informaciones básicas de nuestra red. Aquí nos concentraremos en los problemas específico al NFS.

Como podemos revisar que el servidor remoto se esta ejecutando y bien:

```
# portmap   -v                # Encendemos en modo verbose el port mapper
# rpcinfo  -p  localhost      # $ rpcinfo  -p  nombre_servidor_remoto (sintaxis general)
   program        vers    proto   port
    100000         2      tcp     111     portmapper
    100000         2      udp     111     portmapper
```

Dependiendo en la actividad de la red usted debe ver portmapper, nlockmgr, status, nfs y mountd ejecutándose en TCP y UDP. Puede ser que encuentre varias instancias de nlockmgr y mountd. El nfs mismo sólo

estará ejecutándose sobre UDP (recuerde que es sin estado). Si cualquiera de estos falta, entonces lo más seguro es que el servidor NFS ha fallado en el servidor remoto.

Si recibe muchos timeouts (falla por reintento), puede ser que deba corregir o reajustar incrementando, este parámetro en las opciones de montar de NFS debido a alto tráfico de red. Si esto sucede continuamente, entonces, deberá pensar en otra solución de compartir diferente a NFS ya que esto conlleva a un pobre rendimiento.

Use el comando nfsstat en el servidor:

SALIDA DEL PC AQUI SERVIDOR !

Si badcalls no es igual a ‘0’, entonces la petición RPC están siendo rechazadas por el servidor. Muchas de estos rechazos lo más probable son una indicación de peticiones hechas por el NFS, lo cual significan, casi garantizado, un problema de autentificación.

Use el comando nfsstat en el cliente:

SALIDA DEL PC AQUI CLIENTE !

Si retrans es más de un 5% de las llamadas, entonces las peticiones RPC están dando timeout antes de que el servidor remoto les responda dandole servicio. Debe incrementar el valor de la opción de mount de timeout o reducir la carga al servidor remoto.

Alternativamente, los valores de las opciones rsize y wsize son muy grande. El tamaño por defecto del tamaño de bloque es de 1024, pero este tamaño es muy pesimista y se recomienda un tamaño de 8192. Un tamaño de bloque grande puede significar que los paquetes se están perdiendo. Puede probar con parámetros de digamos 2048, y luego de 4096, y analizar el comportamiento.

RESUMEN

Los siguientes tópicos fueron discutidos en este capítulo:

- NFS permite a sistemas GNU/Linux compartir archivos con otros sistemas, incluyendo sistemas No-GNU.

- Los Clientes usan el comando mount para accesar recursos compartidos. (drive!!!)
- Volumenes que deben ser permanentemente montados se especifican en /etc/fstab.
- El NFS está basado en cuatro protocolos:
    - RPC (Remote Procedure Calls)
    - XDR (External Data Representation)
    - Protocolo NFS
    - Protocolo Mount
- NFS usa UDP para limitar consumo fijo de recurso y permanecer sin estado (stateless).
- NFS tiene debilidades de seguridad que deben ser tomadas en cuenta al implementarse en áreas sensibles.
- Los valores por defecto de NFS pueden ser que necesiten ser ajustados para satisfacer sus necesidades.

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Qué se encuentra en el archivo /etc/exports en el servidor ejecutando NFS?

2.- ¿Cómo funciona el cerrado (locking) de archivos en NFS?

3.- ¿Por qué es necesario el protocolo XDR (External Data Representation)?

4.- ¿Por qué es que el NFS usa UDP?



SAMBA (COMPARTIR ARCHIVOS/IMPRESORAS)

Objetivos

Al finalizar este capítulo, usted estará preparado para efectuar las siguientes tareas:

- Correctamente operar y configurar los daemons de SMB y NMB.
- Implementar la compartición de archivos e impresoras vía Samba.
- Describir la implementación de SMB y NMB
- Listar los programas en la suite de Samba y también sus funciones
- Describir la implementación de Samba y los servicios que los programas proveen, en particular a los usuario.

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son las secciones básicas de los archivos de configuración del servidor de Samba?

2. ¿Cúales programas puede usted usar para conectarse con archivos compartidos por Windows?

3. ¿Cúales daemons necesitan estar siendo ejecutados para que el cliente de samba funcione?

4. ¿Puede ser montado un directorio compartido en Windows ser montado en un sistema GNU/Linux?

5. ¿Cuál archivo contiene la descripción completa de los parámetros de Samba?

INTRODUCCIÓN

En este capítulo discutiremos Samba, el NetBIOS sobre TCP/IP y el administrador de protocolo SMB (Server Message Block) de GNU/Linux. En este capítulo demostraremos como apropiadamente configurar un Servidor Samba y como usar la herramientas clientes para accesar recursos en otros servidores SMB. Estos servidores pueden estar ejecutando sistemas operativos como GNU/Linux, OS/2, Windows, otros UNiXes o programas propietarios suplidos por terceros.

Después de la versión 2.0, Samba también soporta CIFS (Common Internet File System). Este es un esfuerzo financiado por Microsoft para el mejoramiento significativo del protocolo SMB. Las intensiones del desarrollo de CIFS es un sistema de archivos de redes formal y completo, así como es el NFS. Aunque estos proyectos pueden recibir el financiamiento de empresas no totalmente OpenSource o Free Software, Samba es Open Source y distribuido bajo la licencia GPL. Si desea saber más sobre CIFS dirijase al portal de Samba en la dirección web: http://www.samba.org.

SAMBA UN VISTAZO

En esta sección describiremos que es Samba, el protocolo core y los programas que lo conforman, su historia y su uso.

- ¿Qué es Samba?
- Componentes de Samba
- La historia del Samba
- Más allá de Samba

¿Qué es Samba?

Todas las computadoras que usan Windows conectadas a la red se comunican vía el protocolo SMB. Es este protocolo que permite la redirección de archivos de data y trabajos de data de impresión a nodos remota. Los recursos compartidos de discos e impresoras son mediadas por SMB. El SMB esta concebido para ser simultáneamente ambos cliente y servidor. Por esta razón, todas las redes basadas en SMB son realmente peer-to-peer. Esto es un reflejo del origen del protocolo SMB como una solución de compartir recursos de redes basadas en grupos de trabajo.

Similar a los tipos de nombres TCP/IP nombre_host.dominio con DNS y/o archivos de nombre de host para la resolución de direcciones, máquinas ejecutando windows se conocen por un única combinación de workgroup y nombres. Como con los nombres tcp/ip, este par de nombres debe resolver a una dirección de hardware única. El esquema para igualar direcciones de hardware en la red al nombre de un computador y permitir comunicación entre los nodos es conocido como NetBIOS (Network Basic input/output System) y su última extensión, NetBUI (NetBIOS Extended User Interfase).

El NetBIOS y NetBUI fueron desarrollados en la primera parte de los anos 80 paralelo al tcp/ip y fué dirigido específicamente para redes locales, a diferencia del tcp/ip, cual fué diseñado desde el principio para redes de área amplia. Por esta razón, los paquetes NeTBIOS no transportan información de enrutamiento en su cabezal. Para que nodos de NetBIOS puedan comunicarse atraves de segmentos de LAN, el paquete debe ser encapsulado dentro de un protocolo enrutable como es TCP/IP. El protocolo que soporta esta encapsulación NetBT o TCPBEUI y este es el corazón de samba y todo el internetworking de IP de Windows (Hasta windows 2000).

Samba es el software que permite a GNU/Linux, Sistema operativo cual si habla nativamente TCP/IP, interoperar con nodos SMB/NetBIOS. Un tercer nivel de servicio proveído por samba es el Wins (Windows Internet Name Service). El wins trabaja en conjunto con un DNS para proveer resolución de NetBIOS a ip

a direcciones de hardware. Samba puede ser un servidor wins o un cliente wins. Es importante entender que Samba no habla NetBIOS nativamente, sino NBT (NetBIOS sobre TCP/IP). Usted debe tener instalado tcp/ip para usar samba.

Componentes de Samba

Samba consiste de 2 daemons, nmbd y smbd, y un conjunto de programas clientes para accesar los recursos compartidos :

smbclient	La navaja suiza del Samba, Herramienta del lado del cliente.
smbmount	Una versión más sencilla que la anterior para montar los volumenes compartidos.
smbprint	Imprime a un impresora smb, típicamente printers bajo windows.
smbstatus	Muestra las conexiones manejadas por el smbd.
smbpasswd	Maneja las contraseñas de sambas y Windows.
nmblookup	Efectúa consultas de nombres de NetBIOS desde GNU/Linux.
testparms	Verifica el archivo de configuración de Samba.
testprns	Prueba las impresoras definidas en /etc/printcap para el uso con samba.
swat	Un servido de inetd que permite la administración de samba vía web.

El daemon smbd maneja los recursos compartidos y la autentificación de los usuarios. El daemon nmbd es responsable por anunciar y comunicación de los recursos con otras máquinas que entienden el protocolo smb. En la mayoría de los casos estos equipos ejecutan windows pero no tiene que ser así.

Los programas del lado del cliente proveen funcionalidad equivalente al programa en windows de nombre Net.exe. El que sigue es un ejemplo de mapear un disco lógico en windows NT versus samba.

Uso en Windows NT - mapping a drive:
c:\ net use Z:\\e-smith\compartido /USER:ivelis

Uso con Samba - Montando un recurso de la red:
$ smbmount //e-smith/compartido /mnt/compartido -o username=ivelis

La Historia de Samba

La historia de samba se remonta a los primeros días de redes bajo arquitectura PC. En mayo del 85 IBM publico una especificación para una red local de PC's basadas en NetBIOS. Luego apareció el Programa para Redes PC de IBM, con la versión 1.0 del protocolo SMB. Subsecuentemente IBM trabajo con muy cerca con MICROSOFT para desarollar un programa completo de redes peer to peer, cual lanzo para el DOS a finales de los anos 80' a Microsoft LAN Manager, mientras tanto IBM planificaba agregarles las características de redes a sus sistema operativo, OS/2 cual fué diseñado como el reemplazo de DOS. Mucho software fué codificado específicamente a las APIS de NetBIOS asegurandole una larga vida. Microsoft e IBM rompieron sus relaciones debido a la diferencias sobre su proyecto en conjunto el OS/2 microsoft deseaba balancear su mercado existente de DOS con windows, un shell gráfico que se le agregaba al DOS. Mientras que IBM deseaba moverse hacia un sistema operativo totalmente multitarea de 32 bit pero limitado a la compatibilidad del DOS. Esta división también llevo el desarrollo divergente del protocolo SMB. Como windows sobre DOS, se convirtió en el ambiente predilecto, y OS/2 perdió simpatía de sus clientes y desolladores, microsoft se convirtió en la punta de lanza del desarrollo de este protocolo de red que fué inventado por IBM.

Subsecuentemente las mejoras de smb vinieron desde el lado de Microsoft en una serie de lanzamiento de software como son LAN Manager 2.0 a Windows for Workgroups, Windows 95 y finalmente Windows NT. El esquema de autentificación de Windows NT basado en dominios y el esquema de servicios de anuncios conocidos como browsing. Microsoft que no era exactamente muy dada a entregar especificaciones

sobre las operaciones internas de estos productos SMB pero con documentación ya existente y mucho trabajo de ingeniería en reversa, los programadores, en particular Andrew, Tridgell, pudieron armar la operación del smb del producto más reciente y desarollar un software que podía interactuar en redes de windows en una manera completamente transparente este fué el origen del proyecto Samba.

El equipo samba tenia una tarea dificultosa ya que estaban persiguiendo a un objetivo móvil. Microsoft que cada vez parecía menos interesado en compartir información y sus productos de redes. Los lanzamientos de service pack para windows NT han resultado silenciosamente y sin anunciarlos en cambios internos a la operación del smb. Claro esta, hasta que estos cambios son descubiertos y reimplementados en samba, la funcionabilidad de samba dentro de redes de windows puede ser alterada. Pero el equipo de samba se ha manejado brillantemente en mantenerse al compás de estos cambios. Paralelo al kernel de Linux y el servidor apache samba es visto como un de los más importantes esfuerzos de desarrollo Open Source.

Un VISTAZO a SAMBA

Hoy dia samba a llegado a un nivel de excelencia tan elevando que la funcionabilidad de sistemas UNiX y GNU/Linux pueden participar o reemplazar completamente sistemas operativos débiles o menos seguros que comprometen a riesgo de ataques maliciosos y virus. En este momento SAMBA esta en la capacidad de producir el remplazo de un PDC (Primary Domain Controler) de dominio. En la actualidad windows 2000 representa un completa restructuración de las redes de windows pero como windows 2000 es completamente compatible con las redes de windows anteriores entonces también es totalmente compatible con SAMBA. En windows el modelo de dominio va evolucionando y remplazado por un modelo de directorio. Esto impacta fundamentalmente la función del controlador de dominio.

INSTALAR Y CONFIGURAR SAMBA

En esta sección, explicaremos como instalar, configurar y ejecutar SAMBA. Los siguientes tópicos serán discutidos en esta sección:

- Instalar Samba
- Configuración Simple
- Configuración Avanzada
- Configuración de Prueba
- Iniciar el Servidor
- Usuarios de Samba
- Discos Compartidos
- Impresoras Compartidas

Instalar Samba

Instalar Samba casi siempre significa unos cuantos comandos simple de instalación usando el manejador de paquete de su distribución. Existen paquetes disponibles para los 2 manejadores principales. Si usted desea descargar el código fuente y compilarlo en su sistema. Esta puede ser una magnifica idea para un sistema de producción, ya que su compilador recojerá la optimización en particular de su procesador, así asegurandole el mejor desempeño de su software. Usted debe tener soporte para el sistema de archivo SMBFS en su Kernel para poder montar recursos compartidos bajo el protocolo Samba.

Para configurar Samba usted necesita saber las siguiente información:

- El nombre del dominio de Windows NT que el servidor Samba reside. Alternativamente este puede ser el nombre del Workgroup del grupo del trabajo local si Samba participara en un red Peer-to-peer.
- La dirección IP de cualquier servidor WINS en su dominio. En un ambiente de Workgroup, la resolución

de nombres lo más probable será llevada a cabo por el archivo LMHOSTS.

- Los nombres de los usuarios y grupos en el dominio de Windows NT que accesaran los recursos en el servidor Samba para los usuarios de GNU/Linux deseando accesar recursos del dominio deberá crearle cuentas en el dominio de Windows NT.

Configuración Simple

Para configurar un servidor Samba se deben tener en cuenta muchos aspectos. El primero de estos es estar completamente familiarizado con las herramientas que este suite nos hace disponible. Para lograr conocer estas herramientas el primer sitio son las páginas del manual de samba, smbclient, smbmount y smblookup.

SWAT

Existe una herramienta con interfaz Web conocida como SWAT (Samba Web Administration Tool) que facilita el trabajo de configurar a Samba. Esta se instala a través del paquete samba-swat. Es un servicio controlado por el daemon xinetd, su archivo de configuración es /etc/xinetd.d/swat y el puerto por el que funciona por defecto es el 901. Para utilizar SWAT se debe, una vez instalado el paquete necesario, editar el archivo antes mencionado cambiando el atributo disable, cuyo valor es yes por defecto. Por último se reinicializa el servicio xinetd y entonces se hace en un browser que soporte autenticación http://máquina:901. Sino instalo desde un paquete RPM o DEB deberá editar estos dos archivos:

- Agregar la siguiente línea al archivo /etc/services:

```
swat                 901/tcp
```

- Agregar la siguiente línea si usa inet.d o xinet.d:

```
swat  stream  tcp  nowait.400  root  /usr/local/samba/bin/swat  swat
```

Estos comandos agregan a SWAT al archivo /etc/services, cual es usado parra especificar el protocolo y número de puerto que un programa usa. La entrada en el archivo /etc/inet.d.conf o /etc/xinet.d.conf proveerá la información para el demonio inet.d en un caso y xinet.d en el otro. Luego deberemos reiniciar el demonio de inet.d con el siguiente comando:

```
$ killall -HUP inetd
```

Luego de esto la herramienta de SWAT ya está lista para usarse en un navegador de páginas Web. Para conectarse al servidor de Samba en el puerto 901, sólo debe escribir la siguiente dirección en su navegador:

http://localhost:901/

La herramienta SWAT, además de permitir configurar todos los aspectos de Samba, se puede emplear para administrar los usuarios de este servicio, así como conocer las conexiones activas en un momento determinado. También dispone de ayuda. Esta herramienta es una interfaz para evitar la edición manual del archivo de configuración de samba /etc/samba/smb.conf o /etc/smb.conf dependiendo de su distro.

El Archivo de Configuración

La configuración de Samba en un GNU/Linux (u otra máquina UNiX) es controlada por un solo archivo, /etc/smb.conf. Este archivo determina qué recursos del sistema quieres compartir con el mundo exterior y que restricciones deseas poner en ellos.

Como las siguientes secciones ‘direccionarán’ la compartición de unidades e impresoras de GNU/Linux con máquinas Windows, el archivo smb.conf mostrado en esta sección es lo más simple posible, solo para propósitos introductorios.

No te preocupes por los detalles, aún. Otras secciones más adelante introducirán los conceptos más importantes. Cada sección del archivo empieza con una cabecera como [global], [impresoras], etc.

La sección [global] define unas pocas variables que Samba usará para definir la compartición de todos los recursos.

La sección [homes] permite a los usuarios remotos acceder a sus respectivos directorios principales en la máquina Linux local (cada uno al suyo nada más). Esto es, si un usuario de Windows intenta conectar a este recurso desde su máquina Windows, será conectado a su directorio personal. A tener en cuenta que para hacer esto, tiene que tener una cuenta en la máquina GNU/Linux.

El archivo smb.conf que viene debajo como ejemplo permite a los usuarios remotos acceder a su directorio principal en la máquina local y escribir en un directorio temporal. Para que un usuario de Windows vea estos recursos, la máquina Linux debe estar en la red local. Entonces el usuario simplemente conecta una unidad de red desde el Explorador de Windows o el Windows File Manager.

Fíjate que en las siguientes secciones, se darán entradas adicionales a este archivo para permitir la compartición de más recursos.

```
; /etc/smb.conf
;
; Reinicia el servidor cada vez que hagas cambios a este archivo, ej:
; /etc/rc.d/init.d/smb parar
; /etc/rc.d/init.d/smb empezar

[global]
; Quita el comentario a la siguiente línea si quieres cuentas de invitado
; guest account = nobody
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes = yes

[homes]
   comment = Directorios principales
   browseable = no
   read only = no
   create mode = 0750

[tmp]
   comment = Espacio de archivos temporales
   path = /tmp
   read only = no
   public = yes
```

La primera sección es la [global], y todos los parámetros que siguen van en esta sección.

Parámetros Relacionados con la Navegación/Browsing

Browsing es el acto de navegar atraves de la lista de los dominios, grupos de trabajo y nodos que conforman la red NetBIOS (Windows Network). Los parámetros inmediatos relacionados con navegación son el nombre del computador (NetBIOS) y el nombre del grupo de trabajo \dominio. Al menos que sea especificado el nombre del computador se asume ser el mismo que el nombre del host TCP/IP es recomendable que explicitamente cite el nombre NetBIOS, en el archivo smb.conf. El nombre del domino o Workgroup debe ser explícitamente especificado en este archivo smb.conf. Se puede agregar un comentario

que se mostrara en la lista de navegación los 3 parámetros juntos en la sección global deben verse en el siguiente ejemplo:

[global]
netbios name = equipo1
workgroup = dominio
comment = Máquina del peque

Fijese que samba no distingue entre los nombres de dominio y los nombres de workgroup. En una red NetBIOS realmente NetBEUI, un nodo es seleccionado como el navegador maestro (Master Browser). En redes NT este casi siempre es denominado el controlador de dominio el primer nodo el línea naturalmente es el primer master browser. Cuando el segundo nodo entra en línea busca el master browser, y así sucesivamente, eventualmente, cuando un nodo entra puede suceder los siguiente:

- No esta de acuerdo con la elección del master browser
- No puede encontrar el master Browser

En cualquiera de los casos la elección de navegador toma lugar. El nodo NetBIOS, se pone de acuerdo en quien debe manejar la tarea de navegación y ese nodo es designado como el nuevo master browser. Cualquier equipo con una identidad NetBIOS puede funcionar como un master browser. Desde el punto de vista de samba nos queremos asegurar que samba siempre pierda esta elección. Hacemos esto estableciendo el parámetro de nivel de OS en el archivo /etc/smb.conf a 1:

os level = 1

Cada segmento de red tiene un master browser local y existe un master browser a nivel de dominio podemos tener las peticiones nmbd requerir el estado del master browser local nosotros deseamos que samba sea ninguno de estos. Lo que sigue son las entradas necesarias:

local master = no
domain master = no
preferred master = no

Necesitamos decirle a samba que no intente sincronizar la sincronización del navegador atraves de los segmentos de redes y que no se anuncie atraves de los segmentos (le dejamos esta tarea a windows). Logramos esto comentando con un punto y coma (O simplemente eliminandolo) las líneas apropiadas en el archivo de configuración:

; remote browse sync =
; remote announce =

El efecto de todas estas entradas es establecer a samba con un nombre de NetBIOS y un id de grupo de trabajo y entonces para hacerlo completamente pasivo en lo que concierne al tema de navegación.

Ahora necesitamos decirle a samba donde esta el servidor WINS y no queremos que actúe el mismo como un servidor WINS:

wins server = 192.168.2.7 # Reemplace con su wins primario
win support = no # No sea un servidor wins

Luego la próxima tarea es establecer el orden del la resolución de nombre deseamos que el servidor wins maneje la resolución de nombres NetBIOS. Solo así sobrepasamos al DNS o al archivo /etc/host:

name resolve order = host de windows

No utilice el archivo lmhost en dominios NT; El mantenimiento es prohibitivo. WINS es esencial a redes de windows NT y si usted no logra que funcione tendrá un problema mucho mayor que instalar samba. Tampoco dependa de broadcast para resolución de nombre, no puede ser enfatizado lo suficiente - deje que WINS maneje la resolución de nombres y si esto no esta trabajando antes de proceder.

Parámetros Relacionados a los Usuarios y la Seguridad

La próxima tarea es de decirle a Samba como manejar la autentificación de los usuarios. Asumiremos que el Windows NT en la red está utilizando contraseñas encriptadas para autentificar. Usaremos autentificación a nivel de usuario (user-level) y no al nivel de recursos (share-level), ya que este es el método más seguro y más en la línea de la manera que NT funciona al nivel de dominio. Como queremos usar autentificación al user-level y no en el share-level, no permitiremos contraseñas en blanco. Finalmente no queremos la sincronización de contraseñas de Windows NT a GNU/Linux desde el inicio; quizás lo agreguemos Después que todo este funcionando bien más adelante.

Aquí mostramos entradas en el archivo de configuración que muestra lo de arriba discutido:

security = user
encrypt passwords = yes
null passwords = no
smb passwd file = /etc/passwd
unix password sync = no

La seguridad es importante y esta se puede establecer primeramente estableciendo la lista de control de acceso que definirá que máquinas o redes podrán acceder hacia el servidor. El parámetro hosts allow sirve para determinar esto. Si la red consiste en la máquinas con dirección IP desde 192.168.1.1 hasta 192.168.1.254, el rango de direcciones IP que se definirá en hosts allow será 192.168.1. de modo tal que solo se permitirá el acceso dichas máquinas. Note por favor el punto al final de cada rango. Edite ésta de manera que quede del siguiente modo:

hosts allow = 192.168.1. 127.

En esta primera parte no nos interesa restringir el acceso de los hosts, así que comentaremos estas líneas que se refieren a los hosts permitidos (allow) y los denegados (deny):

; allow hosts =
; deny hosts =

Parámetros Relacionados con Redes

El parámetro interfases permite establecer desde que interfases de red del sistema se escucharán peticiones. Samba no responderá a peticiones provenientes desde cualquier interfaz no especificada. Esto es útil cuando Samba se ejecuta en un servidor que sirve también de puerta de enlace para la red local, impidiendo se establezcan conexiones desde fuera de la red local.

interfases = 192.168.1.254/24

Le podemos decir a Samba como manejar la transmisiones TCP (esta opciones son similar a las recomendadas pociones de montar de NFS):

socket options = TCP_NODELAY SO_RECVBUF=8192 S0_SNDBUF=8192

Parámetros Relacionados con Sensitividad de Caso

Aquí le podemos decir a Samba como manejarse con lo que es caso sensitivo o no, para el manejo de las mayúsculas y minúsculas. En este parámetro podemos Especificar como será tratada la búsqueda y salvamento de archivos, ya que Windows es indiferente a las letras (abiertos=AbIeRtOs), mientras que UNiX si toma en cuenta la escritura (abiertos no es Abiertos).

De la manera que Windows NT lo maneja es como le indicamos en este próximo ejemplo, que incluye contraseñas caso sensitivas solamente y en los archivos es caso no sensitivo pero caso preservador:

default case = lower
case sensitive = no
preserve case = yes

password level = 0

La última opción le indica: Usa la contraseña exactamente como se escribió.

Configuración Avanzada

Samba fué creado con un objetivo: ser un reemplazo definitivo para Windows como servidor en una red local. Ésto, por supuesto, requiere algunos procedimientos adicionales dependiendo de las necesidades de la red local. La configuración de Samba involucra más de 230 posible opciones para el archivo /etc/smb.conf. Para cubrir todas estas opciones necesitaríamos todo un libro. Deberá investigar más en las páginas man, libros y los comos (HowTos). En esta sección incluiremos algunos de estos pasos avanzados, del Como Samba de Joel Barrios Dueñas de http://www.linuxparatodos.net, sólo para ilustrar, no se tiene la intención de ser completo.

Re-asignación de grupos de Windows en Samba.

Los grupos que existen en Windows también se utilizan en Samba para ciertas operaciones, principalmente relacionadas con lo que involucra un Controlador Primario de dominio (o PDC que significa Primary Domain Controler). Estos grupos existen de modo predefinido en Samba. Sin embargo, si se ejecuta lo siguiente:

net groupmap list

Devolverá la siguiente información:

System Operators (S-1-5-32-549) -> -1
Domain Admins (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-512) -> -1
Replicators (S-1-5-32-552) -> -1
Guests (S-1-5-32-546) -> -1
Domain Guests (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-514) -> -1
Power Users (S-1-5-32-547) -> -1
Print Operators (S-1-5-32-550) -> -1
Administrators (S-1-5-32-544) -> -1
Account Operators (S-1-5-32-548) -> -1
Domain Users (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-513) -> -1
Backup Operators (S-1-5-32-551) -> -1
Users (S-1-5-32-545) -> -1

Lo anterior corresponde al mapa de los grupos que, de modo predeterminado, utilizará Samba si éste fuese configurado como Controlador Primario de Dominio. XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX corresponde a un número generado aleatoriamente al iniciarse Samba por primera vez. Tome nota de dicho número, ya que lo requerirá más adelante para re-asignar los nombres al español en el mapa de grupos.

Los grupos anteriormente descritos trabajarán perfecta y limpiamente asociándolos contra grupos en el sistema, pero solo si utiliza alguna versión de Windows en ingles. Si utiliza alguna versión de Windows en español, habrá que re-asignar los nombres de los grupos a los correspondientes al español y asociarles a grupos en el sistema, esto a fin de permitir asignar usuarios a dichos grupos y de este modo delegar tareas de administración del mismo modo que en Windows.

Es por tal motivo que si se tiene la intención de configurar Samba como Controlador Primario de Dominio y al mismo tiempo poder hacer uso de los grupos del mismo modo que en Windows, es decir, por mencionar un ejemplo, permitir a ciertos usuarios pertenecer al grupo de administradores del dominio con privilegios de administrador, lo primero será entonces generar los grupos en el sistema ejecutando como root los siguientes mandatos:

groupadd -r administradores

```
groupadd -r admins_dominio
groupadd -r duplicadores
groupadd -r invitados
groupadd -r invs_dominio
groupadd -r opers_copias
groupadd -r opers_cuentas
groupadd -r opers_impresion
groupadd -r opers_sistema
groupadd -r usrs_avanzados
groupadd -r usuarios
groupadd -r usuarios_dominio
```

Una vez creados los grupos en el sistema, solo resta re-asignar los nombres al español en el mapa de grupo de Samba y asociarles a éstos los grupos recién creados en el sistema. El procedimiento se resume a ejecutar algo como lo siguiente:

```
net groupmap modify \
ntgroup=”Nombre grupo Windows en español” \
sid=”número-de-identidad-en-sistema \
unixgroup=grupo_en_linux \
comment=”comentario descriptivo acerca del grupo”
```

Lo anterior establece que se modifique el registro del grupo que corresponda al sid (identidad de sistema) definido con el nombre establecido con ntgroup, asociándolo al grupo en el servidor con unixgroup y añadiendo un comentario descriptivo acerca de dicho grupo con comment.

De modo tal, y a fin de facilitar las cosas a quien haga uso de este manual, puede utilizar el siguiente guión para convertir los nombres al español y asociarlos a grupos en Linux, donde solo deberá definir el número de identidad del sistema que corresponda al servidor:

```
#!/bin/sh
SIDSAMBA=XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX

net groupmap modify ntgroup=”Administradores” \
sid=”S-1-5-32-544” unixgroup=administradores \
comment=”Los administradores tienen acceso completo y sin restricciones al equipo o dominio”

net groupmap modify ntgroup=”Admins. del dominio” \
sid=”S-1-5-21-$SIDSAMBA-512” unixgroup=admins_dominio \
comment=”Administradores designados del dominio”

net groupmap modify ntgroup=”Duplicadores” \
sid=”S-1-5-32-552” unixgroup=duplicadores \
comment=”Pueden duplicar archivos en un dominio”

net groupmap modify ntgroup=”Invitados del dominio” \
sid=”S-1-5-21-$SIDSAMBA-514” unixgroup=invitados \
comment=”Todos los invitados del dominio”

net groupmap modify ntgroup=”Invitados” \
sid=”S-1-5-32-546” unixgroup=invitados \
comment=”Los invitados tienen de modopredeterminado el mismo acceso que los miembros del grupo Usuarios, excep-
to la cuenta Invitado que tiene más restricciones”

net groupmap modify ntgroup=”Operadores de copias” \
sid=”S-1-5-32-551” unixgroup=opers_copias \
comment=”Los operadores de copia pueden sobrescribir restricciones de seguridad con el único propósito de hacer
copias de seguridad o restaurar archivos”
```

**net groupmap modify ntgroup=”Opers. de cuentas” **
**sid=”S-1-5-32-548” unixgroup=opers_cuentas **
comment=”Pueden administrar cuentas de usuarios y grupos del dominio”

**net groupmap modify ntgroup=”Opers. de impresión” **
**sid=”S-1-5-32-550” unixgroup=opers_impresion **
comment=”Pueden operar impresoras del dominio”

**net groupmap modify ntgroup=”Opers. de servidores” **
**sid=”S-1-5-32-549” unixgroup=opers_sistema **
comment=”Pueden administrar sistemas del dominio”

**net groupmap modify ntgroup=”Usuarios avanzados” **
**sid=”S-1-5-32-547” unixgroup=usrs_avanzados **
comment=”Los usuarios avanzados tienen más derechos administrativos con algunas restricciones. De este modo, pueden ejecutar aplicaciones heredadas junto con aplicaciones certificadas”

**net groupmap modify ntgroup=”Usuarios del dominio” **
**sid=”S-1-5-21-$SIDSAMBA-513” unixgroup=usuarios_dominio **
comment=”Todos los usuarios del dominio”

**net groupmap modify ntgroup=”Usuarios” **
**sid=”S-1-5-32-545” unixgroup=usuarios **
comment=”Los usuarios no pueden hacer cambios accidentales o intencionados en el sistema. Pueden ejecutar aplicaciones certificadas, pero no la mayoría de las heredadas”
exit 0

Nota: Este guión en esta incluido en el disco de “Extras de curso” de Linux Para Todos. Solo basta editarlo y definir la variable SIDSAMBA y ejecutarlo como root.

Una vez hecho lo anterior, al volver a ejecutar lo siguiente:
net groupmap list

Se deberá de mostrar ahora esto otro:
Opers. de servidores (S-1-5-32-549) -> opers_sistema
Admins. del dominio (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-512) -> admins_dominio
Duplicadores (S-1-5-32-552) -> duplicadores
Invitados (S-1-5-32-546) -> invitados
Invitados del dominio (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-514) -> invitados
Usuarios avanzados (S-1-5-32-547) -> usrs_avanzados
Opers. de impresión (S-1-5-32-550) -> opers_impresion
Administradores (S-1-5-32-544) -> administradores
Opers. de cuentas (S-1-5-32-548) -> opers_cuentas
Usuarios del dominio (S-1-5-21-XXXXXXXXXX-XXXXXXXXXX-XXXXXXXXXX-513) -> usuarios_dominio
Operadores de copias (S-1-5-32-551) -> opers_copias
Usuarios (S-1-5-32-545) -> usuarios

De este modo, si por ejemplo, se agrega al usuario fulano al grupo admins_dominio, se tendrá el mismo efecto que si se hiciera lo mismo en Windows agregando al usuario al grupo Admins. del dominio. Esto por supuesto solamente tendrá utilidad si Samba se configura y utiliza como Controlador Primario de Dominio.

Alta de cuentas de usuario en Controlador Primario de Dominio

Si se configuró Samba para funcionar como Controlador Primario de Dominio, será necesario asignar a root una clave de acceso en Samba, la cual por supuesto puede ser diferente a la del sistema, debido a que las

estaciones de trabajo necesitan autenticar primero con el usuario root de Samba para poder unirse dominio y poder crear de este modo una cuenta de máquina en el sistema a través del parámetro add machine script ya descrito anteriormente.

Los usuarios es necesario darlos de alta de modo que queden agregados a los que correspondan en el sistema a grupos Usuarios y Usuarios del dominio de Windows, es decir a los grupos usuarios y usuarios_dominio.

```
useradd -s /sbin/nologin -G usuarios,usuarios_dominio usuario-windows
smbpasswd -a usuario-windows
```

Si el usuario ya existiese, solo será necesario agregarlo a los grupos usuarios y usuarios_dominio con gpassswd del siguiente modo:

```
gpasswd -a usuario-windows usuarios
gpasswd -a usuario-windows usuarios_dominio
```

En teoría en el directorio definido para el recurso Profiles se deben crear automáticamente los directorios de los usuarios donde se almacenarán los perfiles. De ser necesario es posible generar éstos directorios utilizando el siguiente guión:

```
cd /home
for user in *
do
mkdir -p /var/lib/samba/profiles/$user
chown $user.$user /var/lib/samba/profiles/$user
done
```

Parámetros de configuración avanzada en el archivo smb.conf

Anunciando el servidor Samba en los grupos de trabajo

La opción remote announce se encarga de que el daemon nmbd se anuncie a si mismo de forma periódica hacia una red en particular y un grupo de trabajo específico. Esto es particularmente útil si se necesita que el servidor Samba aparezca no solo en el grupo de trabajo al que pertenece sino también otros grupos de trabajo. El grupo de trabajo de destino puede estar en donde sea mientras exista una ruta y sea posible la transmisión exitosa de paquetes.

```
remote announce = 192.168.1.255/MI-DOMINIO 192.168.2.255/OTRO-DOMINIO
```

El ejemplo anterior definió que el servidor Samba se anuncie a si mismo al los grupos de trabajo MI-DOMINIO y OTRO-DOMINIO en las redes cuyas IP de transmisión son 192.168.1.255 y 192.168.2.255 correspondientemente.

Ocultando y denegando acceso a ficheros

No es conveniente que los usuarios acceder o bien puedan ver la presencia de ficheros ocultos en el sistema, es decir ficheros cuyo nombre comienza con un punto, particularmente si acceden a su directorio personal en el servidor Samba (.bashrc, .bash_profile, .bash_history, etc.). Puede utilizarse el parámetro hide dot files para mantenerlos ocultos.

```
hide dot files = Yes
```

En algunos casos puede ser necesario denegar el acceso a cierto tipo de ficheros del sistema. El parámetro veto files se utiliza para especificar la lista, separada por diagonales, de aquellas cadenas de texto que denegarán el acceso a los ficheros cuyos nombres contengan estas cadenas. En el siguiente ejemplo, se denegará el acceso hacia los ficheros cuyos nombres incluyan la palabra «Security» y los que tengan extensión o terminen en «.tmp»:

veto files = /*Security*/*.tmp/

Opciones para cliente o servidor Wins

Puede habilitar convertirse en servidor WINS o bien utilizar un servidor WINS ya existente. Se puede ser un servidor WINS o un cliente WINS, pero no ambas cosas a al vez.

Si se va ser el servidor WINS, debe habilitarse lo siguiente:

wins support = Yes

Si se va a utilizar un servidor WINS ya existente, debe quitar el comentario de la siguiente línea y especificar que dirección IP utiliza dicho servidor WINS:

wins server = 192.168.1.1

Opciones específicas para Controlador Primario de Dominio (PDC)

Si se va a configurar Samba como Controlador Primario de Dominio, se debe especificar todos los parámetros descritos a continuación. Si se quiere que las claves de acceso del sistema y Windows se mantengan sincronizadas, es necesario descomentar las siguiente líneas:

unix password sync = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
passwd:*all*authentication*tokens*updated*successfully

El parámetro local master define al servidor como examinador del dominio (o master browser); El parámetro domain master define al servidor maestro del dominio; El parámetro preferred master define al servidor como maestro del domino preferido en caso de haber más servidores presentes en el mismo dominio como controladores de dominio; El parámetro time server se utiliza para definir que las estaciones deberán sincronizar la hora con el servidor al unirse al dominio; El parámetro domain logons define que el servidor permitirá a las estaciones autenticar contra Samba.

local master = Yes
domain master = Yes
preferred master = Yes
time server = Yes
domain logons = Yes

La configuración de Controlador Primario de Dominio requiere además definir donde se almacenarán los perfiles de los usuarios. Windows 95, 98 y ME requieren se defina con el parámetro logon home, en tanto que Windows NT, 2000 y XP requieren se haga con el parámetro logon path. Para efectos prácticos y de previsión, utilice ambos parámetros y defina la unidad H para dicho volumen:

logon path = \\%L\Profiles\%U
logon home = \\%L\%U\.profile
logon drive = H:

Si se va a utilizar Samba como Controlador Primario de Dominio, es necesario establecer el guión que ejecutarán las estaciones Windows al conectarse hacia el servidor. Esto se hace a través del parámetro logon script el cual puede definir o bien un guión a utilizar por cada usuario (%u.bat) o bien por cada máquina (%m.bat) o bien de modo general para todos (logon.cmd). Para no complicar las cosas, defina inicialmente un guión general para todos del siguiente modo:

logon script = logon.cmd

El fichero /var/lib/samba/netlogon/logon.cmd deberá contener algo como lo siguiente:

REM windows client logon script

```
REM

net time \\mi-servidor /SET /YES
net use H: \\mi-servidor\homes /PERSISTENT:NO
```

El Controlador Primario de Dominio va a necesitar también se definan los guiones a ejecutar para distintas tareas como alta de máquinas, usuarios y grupos así como la baja de estos.

```
add user script = /usr/sbin/useradd %u
add machine script = /usr/sbin/useradd -d /dev/null -g 100 -s /bin/false -c "Cuenta de máquina" -M %u
delete user script = /usr/sbin/userdel %u
delete group script = /usr/sbin/groupdel %g
add user to group script = /usr/bin/gpasswd -a %u %g
set primary group script = /usr/sbin/usermod -g %g %u
```

El parámetro add user script sirve para definir lo que se deberá ejecutar en el trasfondo en el sistema para crear una nueva cuenta de usuario. El parámetro add machine script es particularmente importante porque es el mandato utilizado para dar de alta cuentas de máquinas (trust accounts o cuentas de confianza) de modo automático. El parámetro delete user script es para definir lo propio para eliminar usuarios, delete group script para eliminar grupos, add user to group para añadir usuarios a grupos y set primary group script para establecer un grupo como el principal para un usuario.

Directorio para Netlogon y perfiles en Controlador Primario de Dominio (PDC).

Si se va a utilizar Samba como Controlador Primario de Dominio, es necesario definir los recursos donde residirá netlogon y también donde se almacenarán los perfiles de los usuarios:

```
[netlogon]
 comment = Network Logon Service
 path = /var/lib/samba/netlogon
 write list = @administradores, @admins_abiertos
 guest ok = Yes
 browseable = Yes

[Profiles]
 path = /var/lib/samba/profiles
 read only = No
 guest ok = Yes
 create mask = 0600
 directory mask = 0700
```

Genere con el mandato mkdir los directorios /var/lib/samba/profiles y /var/lib/samba/netlogon. El directorio /var/lib/samba/profiles deberá pertenecer a root y al grupo users y tener permiso 1777 a fin de permitir crear el directorio de perfil correspondiente para cada usuario.

```
mkdir -p -m 1777 /var/lib/samba/profile
mkdir -p /var/lib/samba/netlogon
chgrp users /var/lib/samba/profile
```

Probar la Configuración

El programa de testparm lee el archivo de configuración y reporta cualquier error de sintaxis. Cualquier parámetro de configuración aún no establecido será ajustado a su valor por defecto. La salida puede ser redireccionada a un archivo. Esta salida contiene todos los parámetros conocidos a testparm, las cuales están disponibles para smbd y nmbd. También debe entender que testparm solo revisa sintaxis, no el contenido. Recuerde que su archivo de configuración puede estar sintacticamente correcto pero no apropiado para su ambiente.

```
$ testparm                          # chequea la configuración general del servidor Samba.
$ testparm ivelis 192.168.168.42    # permite chequear los niveles de acceso que tiene el host
                                    # especificado a cada uno de los recursos compartidos. Se
                                    # deben indicar tanto el nombre NetBios como la dirección
                                    # IP de la máquina
```

Iniciar el Servidor

Para poder poner Samba a trabajar, es necesario que inicie a ambos smbd y nmbd. Una vez ejecutándose, estos daemons muy rara vez tendrán que reiniciarse, ya que ellos vuelven y leen el archivo de configuración cada vez que se efectúa una petición. Como se inician los daemons depende en que tipo de instalación utilizo. Por lo general usaría un script en el directorio /etc/rc.d/init.d llamado samba o smb.

En sistemas RedHat y sus distros compatibles como Mandriva, usted puede controlar el servidor con el siguiente comando:

$ /etc/rc.d/init.d/smb opción #opción es una de: start, restart o stop

Si surgen problemas puede revisar los archivos logs en /var/log/samba/log.smb y log.nmb para asegurarse de que están funcionando apropiadamente.

Si iniciará Samba por primera vez ejecute lo siguiente:

/sbin/service smb start

Si va a reiniciar el servicio, ejecute lo siguiente:

/sbin/service smb restart

Para que Samba inicie automáticamente cada vez que inicie el servidor solo ejecute el siguiente mandato:

/sbin/chkconfig smb on

Usuarios de Samba

Al definir usuarios en para Samba, el punto clave es mantener en mente que los usuarios de Samba también serán usuarios de dominios de o locales (usuarios definidos localmente en computadoras ejecutando NT) de Windows NT. así que la tarea principal es la de mapear los usuarios de Windows NT a los de GNU/Linux. Esto se hace atraves de dos archivos, el smbpasswd y el smbusers, ambos se encuentran en el directorio /etc.

La mejor herramienta para agregar usuarios a Samba es el script escrito en Perl smbadduser. Este script debe estar en /usr/bin. Algunas distros no incluyen este archivo; si la suya es el caso entonces descarguelo desde el Internet. El sintaxis de este comando smbadduser es:

$ smbadduser linuxid:NTid

Le preguntará dos veces la contraseña para el nuevo usuario de Samba. Fijese que tanto el nuevo usuario de GNU/Linux como el de NT deben existir.

Los usuarios de Windows NT se asume implícitamente que pertenecen a;l dominio especificado en la entrada de workgroup en el archivo /etc/smb.conf. Para agregar usuarios de otro dominio o de una máquina de NT en particular, especifique el nombre del dominio o el nombre de la máquina explícitamente en el formato dominio/nombre_usuario (o nombre_máquina/nombre_usuario).

El comando smbadduser actualiza los archivos /etc/smbpasswd y el archivo /etc/smbusers de los usuarios de Samba.

Para eliminar un usuario, simplemente edite y elimine las líneas correspondientes en los archivos smbusers y smbpasswd. Otra característica a señalar es que podemos mapear cualquier número de usuarios de NT a un sólo usuario de GNU/Linux. La ventaja que esto ofrece es que podemos definir un usuario de GNU/Linux con permisos restrictivos, entonces mapeamos un grupo por completo de NT a este sólo usuario.

Ejercicios 11-1: Defina Usuarios de Samba

Debe tener una ID de usuario válida en un dominio de Windows NT. La soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Asegúrese que el archivo /etc/smb.conf esta correctamente establecido para su dominio NT y que Samba se esta ejecutado.

2.- Use el comando smbadduser para mapear el ID de un usuario en un dominio de Windows NT al equipo local de GNU/Linux usando el mismo nombre de usuario (ahí es necesario deberá crear el usuario).

Compartiendo Discos

Bueno ya tenemos una configuración válida de Samba y algunos usuarios de Samba, podemos empezar a definir un recurso que queremos compartir en el archivo de configuración /etc/smb.conf. Este tipo de recurso tiene el siguiente formato en su entrada en el archivo:

```
[Disco_USB]
comment = Una definición de su recurso
path = /algun_directorio
guest ok = yes
browseable = yes
writeable = yes
read list = nombres_usuarios
write list = nombres_usuarios
admin list = nombres_usuarios
```

Esto define un recurso compartido de nombre Disco_USB que esta escribible por nombres_usuarios (recuerde que estos son nombres de usuario de GNU/Linux) y se puede browsear desde el Network Neighborhood. Como todo lo de Samba existen muchas variaciones más para crear los recursos compartidos.

Cuando intente conectarse a este recurso desde Windows, recibirá un prompt de dialogo preguntandole por un nombre de usuario y contraseña. Si el nombre de usuario no iguala uno de los usuarios definidos en la lista, la autentificación con los permisos correspondiente (read, write, admin) será denegada, sin importar si el usuario en realidad tiene permisos o no en la máquina ejecutando GNU/Linux. Esto proveé un nivel más de seguridad de autenticación para los compartidos de discos.

Compartiendo Impresoras

Para compartir impresoras también es necesario editar el archivo /etc/smb.conf. Hay dos secciones en el archivo que son de interés aquí: la sección [global], donde se describe toda la configuración del sistema y la sección de cada recurso. Ahora le presentamos un ejemplo de que puede aparecer en la sección [global]. Esta define cual archivo de configuración de impresora (/etc/printcap por defecto) es de ser usado, que estilo imprimir y un valor booleano describiendo si el printer debe ser cargado o no.

```
printin = bsd
printcap name = /etc/printcap
load printers = yes
```

En este ejemplo anterior, el printcap es usado para leer cuales impresoras serán compartidas en el sistemas. Si una sección se llama [Impresoras] se encuentra en el archivo de configuración, entonces el printer lis-

tado en el archivo printcap será desplegado.

En el siguiente ejemplo ilustramos una definición típica de un printer:
[printers]
comment = Todos las Impresoras
browseable = no
printable = yes
public = no
read only = yes
create mode = 0700
directory = /tmp

Algo a tener muy en cuenta es que para poder imprimir en una impresora de Samba, primero tiene que tener su printer de UNiX estos pueden ser del tipo BSD, LPRNG, SYSV y sobre una máquina ejecutando GNU/Linux. El Samba enruta toda la impresión por defecto hacia LPD. Esto se específica en el archivo de configuración smb.conf así “printing =bsd”. este aspectos del Samba puede ser utilizado para simplificar la expresión remota de GNU/Linux sin importar si necesita la conectividad de Windows ya que es mucho más simple trabajar con impresoras SMB compartidas que el dolor de cabeza del Lpd que Samba maneja automáticamente.

Fijese que las impresoras pueden ser configurada como impresoras remotas de smb con la herramienta printtool. Esta es la manera más natural de configurar una impresora remota de Windows.

En la gran mayoría de servidores no es aconsejable compartir todos los recursos de impresión disponible en el servidor. así que, ciertas técnicas de administración deben ser implementadas para controlar o eliminar el acceso a impresoras en específico. Existen dos métodos para prevenir que una impresora sean compartidas el primero es explícitamente describir que la impresora no deberá ser compartida y establecer la opción entodas ellas “printable=no”. El otro método es crear un archivo especial printcap y colocarlo en solo esas impresoras que van ha ser compartidas

LA OPERACION DEL CLIENTE SAMBA

Lo que hemos descrito hasta este momento involucra establecer a Samba como un servidor de archivo e impresión. Aquí nosotros queremos usar a GNU/Linux como un cliente en una red de Windows.Esto es mucho más simple que lo anterior. Aquí describiremos las dos tareas principales: Conectarse a un volumen compartido de Windows y conectarse a una impresora de Windows.

Observe que la naturaleza de la impresión en UNiX no nos deja aprovechar el soporte de impresión bajo Windows. En particular no podemos utilizar las opciones disponible a las impresoras conectadas a las estaciones de trabajo de Windows 98 o NT. Tenemos que enviar las peticiones de impresión atraves de los filtros locales de impresión igual que el estilo de impresión de BSD. Esto no es tan critico ya que todos los días más y más fabricantes producen manejadoras de impresión para sistemas GUN/Linux , hasta el punto de que linux de han ido ganando el espacio como servidor de impresión reensamblando a windows y Novel en los centros de trabajo.

Los siguientes tópicos serán discutido en esta sección:
- Programas Cliente
- Conectarse a Recursos Compartidos de Windows

- Conectarse a Impresoras de Windows

Programas Cliente

Los programas clientes principal de Samba son: smbclient, smbmount, nmblookup y smbfind. Estas herramientas se ejecutan bajo línea de comandos pero existen otras herramientas, y muchos front ends o interfases gráficas para manejar las funciones de Samba, no podemos en el marco de este libro cubrirlas todas, así es que se las dejamos de tarea. En esta sección le hablaremos de estas cuatro más importantes.

smbclient

El comando smbclient permite acceder a recursos compartidos por un servidor SMB con una interfaz similar a la de los clientes FTP, estableciendo una conexión con dicho servidor. Una vez establecida la conexión SMB, se pueden trasladar, borrar e imprimir ficheros, entre otras funcionalidades.

Mediante el empleo de smbclient también es posible hacer consultas de lo compartido a través del protocolo por un servidor determinado, además de poder utilizarce para enviar mensajes mediante el protocolo WinPopup. Su Sintaxis es:

smbclient [servicio] [password] [opciones]

- La opción servicio indica cuando se quiere establecer una conexión SMB con un determinado recurso compartido. Toma la forma //servidor/recurso o podemos identificarlo por su IP. Donde servidor es el nombre NetBios del servidor y recurso es el nombre de lo compartido por dicho servidor. Para resolver el nombre del servidor se pueden emplear varios métodos en un determinado orden. Dichos métodos son mencionados más adelante. Dos ejemplos son:
  - **//estudiante8/temp**
  - **//192.168.168.72/printer**

- La opción password es la contraseña que se utiliza para conectarse al recurso compartido. No existe una contraseña por defecto por lo que de no especificarse, el cliente siempre la solicitará aunque el recurso a acceder no la requiera. En este caso se presionará ENTER para indicar un password vacío. Con la opción -N, descrita más adelante, se puede evitar la pregunta.

- La opción opciones son las que describen como se accede al recurso o que indican otra acción a realizar. Algunas opciones:
  - -U <usuario> : permite especificar el usuario con el que se quiere establecer la conexión. En su defecto se utiliza el valor de la variable del entorno USER, y si esta no tiene valor se emplea el usuario GUEST. Solo es importante su utilización cuando se accede a recursos restringidos por usuario, por ejemplo para los servidores Windows NT.
  - -L <host> : permite consultar a un host servidor del protocolo acerca de todos los recursos que comparte. Además muestra los dominios o grupos de trabajo accesibles para este, así como el PDC5.1 de cada uno de los dominios. Para referirse al host se puede emplear su nombre Netbios o número IP. En este caso no se específica ningún servicio.
  - -N : evita que se pregunte el password al establecer la conexión.
  - -I : permite especificar la dirección IP del servidor. Es útil cuando el mecanismo de resolución de nombres no devuelve la dirección deseada.
  - -M <host> : permite enviar mensajes a través del protocolo WinPopup a un <host> que tenga activado el servicio. El contenido del mensaje se toma de la entrada estándar durante la ejecución del comando.
  - -R <orden de resolución de nombres> : específica que mecanismos se utilizarán para resolver los nombres y en que orden se aplicarán. Los posibles mecanismos son:
    - -lmhosts : se resuelve el nombre mediante el fichero lmhosts de Samba instalado mediante el paquete samba-common en el directorio /etc/samba. Su formato es similar al de Windows.
    - -host : se resuelve el nombre mediante los mecanismos tradicionales de UNiX y Linux: fichero de hosts (/etc/hosts), NIS y DNS (ver el capítulo http://www.linux.cu/manual/avanzado-html/node56.html - DNS, pág. http://www.linux.cu/manual/avanzado-html/node56.html - DNS).

- -wins : se resuelve el nombre mediante un servidor WINS. Este se específica mediante el fichero de configuración de Samba, /etc/samba/smb.conf descrito más adelante.
- -bcast : se resuelve el nombre mediante un paquete broadcast a todas las máquinas en la misma subred de todas las interfases de red disponibles y configuradas en smb.conf.

El orden en que se emplean estos mecanismos por defecto es lmhosts host wins bcast. Este se puede variar mediante la opción o la configuración en smb.conf

Una vez establecida la conexión SMB se muestra un prompt en el cual se pueden ejecutar, entre otros, los siguientes comandos:

- help [comando] : muestra los posibles comandos a ejecutar o la utilización de un comando si se específica. También puede usarse ?.
- ls [patrón] y dir [patrón] : listan los ficheros del directorio actual en el servidor que satisfagan el patrón especificado como argumento. Por defecto se asume el patrón ``*''.
- get <fichero_remoto> [fichero_local] : transfiere un fichero desde el servidor al cliente salvándolo opcionalmente con el nombre especificado en el segundo argumento.
- mget <patrón> : transfiere desde el servidor al cliente todos los ficheros que satisfagan el patrón especificado.
- put <fichero_local> [fichero_remoto] : transfiere un fichero desde el cliente al servidor salvándolo opcionalmente con el nombre especificado en el segundo argumento.
- mput <patrón> : transfiere desde el cliente al servidor todos los ficheros que satisfagan el patrón especificado.
- recurse : activa y desactiva la transferencia recursiva de directorios para los comandos mget y mput. También determina que la salida de los comandos ls y dir sea recursiva o no.
- prompt : activa y desactiva el modo interactivo al hacer las transferencias múltiples con mget y mput.
- rm, rd y rmdir : permiten borrar ficheros y directorios en el servidor.
- !<comando> : ejecuta un comando en un shell en el cliente.
- print <fichero> : imprime un fichero si se está conectado a una impresora.
- queue : si se está conectado a una impresora, imprime la cola de impresión de esta.
- exit y quit : cierran la conexión con el servidor.

Algunos ejemplos son:

```
# smbclient //PROXY-AP/Primary -U admin
added interfase ip=127.0.0.1 bcast=127.255.255.255 nmask=255.0.0.0
added interfase ip=10.0.0.1 bcast=10.0.0.255 nmask=255.255.255.0
Got a positive name query response from 127.0.0.1 ( 10.0.0.1 )
Password:
Domain=[MITEL-NETWORKS] OS=[UNiX] Server=[Samba 2.2.8a]
smb: \>
```

```
# smbclient -L 10.0.0.1 -R bcast -N
added interfase ip=127.0.0.1 bcast=127.255.255.255 nmask=255.0.0.0
added interfase ip=10.0.0.1 bcast=10.0.0.255 nmask=255.255.255.0
Anonymous login successful
Domain=[MITEL-NETWORKS] OS=[UNiX] Server=[Samba 2.2.8a]

        Sharename       Type            Comment
        ---------       ----            -------
        print$          Disk            Printer drivers
        Primary         Disk            Primary i-bay
        IPC$            IPC             IPC Service (Mitel Networks SME Server)
        ADMIN$          Disk            IPC Service (Mitel Networks SME Server)

        Server                          Comment
        ---------                       -------
        PROXY-AP                        Mitel Networks SME Server

        Workgroup                       Master
```

```
---------                 -------
IMPRESORA                 ABIERTOS
MITEL-NETWORKS            PROXY-AP
```

$ smbclient //PROXY-AP/IMPRESORA -I 192.168.168.4

$ smbclient -M PROXY-AP
added interfase ip=192.168.168.89 bcast=192.168.168.255 nmask=255.255.255.0
Got a positive name query response from 192.168.168.4 (192.168.168.4)
Connected. Type your message, ending it with a Control-D
Hola. Probando el WinPopup
Ctrl-d
sent 13 bytes

$ cat message | smbclient -M ivelis # toma el contenido del archivo message y lo envía a ivelis

smbmount

El comando smbmount permite montar un recurso compartido utilizando el protocolo SMB, en el sistema de ficheros local. Para ello el kernel debe soportar el file system SMB. El sintaxis es:

smbmount <servicio> <directorio> [-o <opciones>]

El servicio se específica de la misma forma que en el comando smbclient y las opciones se indican en forma de pares <opción>[=<valor>] y se separan por comas. Algunas opciones son:

- username : permite indicar el nombre del usuario con el que se accede al recurso.
- password : permite indicar el password del usuario con el que se accede al recurso.
- fmask : permite indicar los permisos que tomarán los ficheros (en formato numérico) en el directorio montado. De no especificarse se utiliza la máscara del shell actual.
- dmask : es similar a fmask pero para los directorios.
- gid y uid : permiten indicar los identificadores de grupo y de usuario respectivamente, asignados al recurso montado. Por defecto se asumen el ID del usuario y del grupo actuales (uid y gid).
- rw y ro : indican si el recurso se monta para lectura y escritura, o para lectura solamente.

Este comando sólo puede ser utilizado por los usuarios distintos de root si se le asigna el permiso especial ``s'' ejecutando:
chmod +s /usr/bin/smbmount

Ejemplo:
$ smbmount //PROXY-AP/temp /mnt/temp -o username=ivelis,fmask=700,ro
Password:
Para desmontar un recurso previamente montado con smbmount, se emplea el comando smbumount que también debe tener asociado el permiso ``s'' para ser ejecutado por usuarios distintos de root.

Ejemplo: $ smbumount /mnt/temp

nmblookup

El comando nmblookup básicamente permite hacer consultas acerca de los nombres NetBios en una subred. Sin opciones, el comando traduce un nombre NetBios a dirección IP. El sintaxis es:

nmblookup [opciones] <nombre_netbios>

Algunas opciones:

-A :
Indica que se interprete el argumento como una dirección IP devolviendo el estado del nodo del host correspondiente, o sea todos los nombres NetBios registrados por dicho host.

-T :
Indica que se realice una consulta inversa al DNS a partir del número IP obtenido.

-M :
Indica que se busque el ``Master Browser'' del nombre NetBios dado como argumento, para ello utiliza el tipo de nodo 0x1d.

-S :
Indica que además de devolverse la dirección IP se muestre el estado del nodo correspondiente, o sea todos los nombres NetBios registrados por el host.

Para expresar el tipo de nombre NetBios se puede añadir al nombre principal una cadena de la forma ``#<tipo>''. También en lugar de un nombre NetBios, como argumento se puede colocar la cadena '*' (incluyendo las comillas), lo cual indicará que se consulte a todos los hosts alcanzables mediante un paquete broadcast en todas las interfases de red configuradas.

Aquí le presentamos varios ejemplos son:

```
$ nmblookup equipo1
querying equipo1 on 192.168.168.255
192.168.168.42 equipo1<00>

$ nmblookup '*'
192.168.168.102 *<00>
192.168.168.72 *<00>
192.168.168.103 *<00>
192.168.168.214 *<00>
...

$ nmblookup -S -T PROXY-AP
querying PROXY-AP on 192.168.168.255
PROXY-AP.abiertos.net, 192.168.168.30 PROXY-AP<00>
Looking up status of 192.168.168.30
    PROXY-AP        <00> -          B <ACTIVE>
    PROXY-AP        <03> -          B <ACTIVE>
    PROXY-AP        <20> -          B <ACTIVE>
    DESARROLLO      <00> - <GROUP> B <ACTIVE>
    DESARROLLO      <1e> - <GROUP> B <ACTIVE>

$ nmblookup DESARROLLO#1e
querying DESARROLLO on 192.168.100.255
192.168.168.102 DESARROLLO<1e>
192.168.168.214 DESARROLLO<1e>
192.168.168.31  DESARROLLO<1e>
192.168.168.33  DESARROLLO<1e>
...
```

smbfind

El comando findsmb es un script de perl que lista información acerca de todas las máquinas en una subred que responden al protocolo SMB. Internamente emplea al comando nmblookup. Para que devuelva una información más completa debe ejecutarse como root y no debe estar ejecutándose el servicio nmbd.

Como argumento solo recibe la dirección de broadcast a utilizar, asumiendo la correspondiente a la interfaz de red empleada.

Aquí le presentamos un ejemplo:

```
# findsmb
IP ADDR          NETBIOS NAME   WORKGROUP/OS/VERSION
---------------------------------------------------------------------
192.168.168.1     MAILDI         *[DISAIC.NET] [Windows NT 4.0] [NT LAN Manager 4.0]
192.168.168.31    GLOIN           [AULA] [UNiX] [Samba 2.2.2]
192.168.168.77    PROXYDI         [DISAIC.NET] [Windows 5.0] [Windows 2000 LAN Manager]
192.168.168.79    DELTHA          [DISAIC.NET]
192.168.168.102   SION            [AULA] [UNiX] [Samba 2.2.1a]
192.168.168.145   MERLIN         +[DIR.GRAL]
192.168.168.203   VENUS           [DISAIC.NET] [Windows NT 4.0] [NT LAN Manager 4.0]
...
```

Existen otros comandos como smbprint y smbtar que son shell scripts basados en smbclient. El primero permite hacer transparente la impresión a través de una impresora compartida por un servidor SMB y previamente configurada (se puede hacer con printtool). El segundo se utiliza para hacer copias de seguridad de ficheros en el servidor SMB en un dispositivo local (backups).

Conectarse a Recursos Compartidos de Windows

Hay dos maneras de conectar un sistemas GNU/Linux a un recurso compartido SMB. Usted puede usar el smbclient para recibir información a cerca de un recurso compartido (share) y para conectarse a ella en una manera parecida al FTP. El método más común es montar el share en el sistema GNU/Linux usando el comando smbmount. Sólo discutiremos el segundo método. Recuerde que su kernel debe tener compilado el módulo de soporte para el sistema de archivos smbfs.

Montar un recurso compartido SMB es muy parecido a montar un sistema de archivos NFS, con algunos cambios de sintaxis:

$ smbmount //nombre_servidor/Compartido /ruta/local -o opciones

Fijese que en los nombres de compartidos las barras (/) son reemplazadas por una barra doble (//) al principio del nombre del share.

La opción principal desde el punto de vista del usuario son username, password y workgroup. El workgroup es asumido desde el archivo .etc.smb.conf al menos que sea explícitamente especificado. La contraseña se transporta en texto plano, lo que es un serio riesgo de seguridad. Otro ejemplo del uso típico del comando smbmount:

$ smbmount //nombre_servidor/Compartido /ruta/local -o username=admin password=admin123

Conectarse a Impresoras de Windows

La herramienta printtool define la pila (el stack) de impresión que se conecta a una impresora remota de Windows. Primero usamos el comando smbclient para descubrir el nombre de la impresora como Samba lo ve. Si aún esta usando BSD para imprimir estará limitando la base de datos de impresoras disponibles. No espere encontrar una gran cantidad de filtros, y tendrá muchs veces que conformarse con utilizar uno

que se acerca al suyo. Si necesita utilizar las características avanzadas de su impresora moderna desde GNU/Linux, deberá instalar un sistema de impresión mucho más capaz como es el recomendado CUPS.

Ejercicios 11-2: Habilitar swat

Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Verifique que el servicio de inetd/xinetd este configurado para el puerto 901.

2.- Edite el archivo /etc/inetd.conf y elimine el comentario listando el servicio de swat.

3.- Reinicie el daemon de inetd para habilitar el servicio de swat.

4.- Entre al sistema X Window e inicie un navegador web. Escriba la url http://localhost:901/ y entre a la administración de de Samba atraves de swat.

Ejercicios 11-3: Crear un Recurso Compartido de Samba

Para poder llevar acabo este ejercicio es necesario que ya tenga un servidor de samba ejecutándose y una estación de trabajo ejecutando Windows (98 o NT). Si ha seguido todos los pasos hasta el ejercicio anterior entonces ya este debe ser el caso. Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Cree un recurso compartido y llamelo Compartidos apuntelo hacia el directorio /home/usuario/mis_documentos. El nombre del usuario debe ser el mismo usuario que el de la estación de trabajo de Windows NT. Otorguese permisos administrativos a este recurso. El recurso de establecerlo como escribible/writeable y navegable/browseable. Recuerde que en el ejercicios 11-1 se creo este usuario en la estación de trabajo GNU/Linux.

2.- Use el utilitario smbclient en el servidor de Samba para verificar que el recurso está listado efectivamente en los servicios disponibles.

3.- Dirijase a la estación de trabajo ejecutando Windows e ingrese (login) con el usuario que se creo y al cual le compartió el recurso. Use la aplicación gráfica Network Neighborhood para navegar el servidor de GNU/Linux. Verifique que puede navegar e escribir remotamente al recurso compartido.

Ejercicios 11-4: Montar un Recurso Compartido de Windows

Para poder llevar acabo este ejercicio es necesario que ya tenga un servidor de samba ejecutándose y una estación de trabajo ejecutando Windows (98 o NT). Si ha seguido todos los pasos hasta el ejercicio anterior entonces ya este debe ser el caso. Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Encuentre un recurso compartido en la estación de trabajo de Windows (NT o 98) que usted tenga otorgado permisos de escritura.

2.- Usando el comando smbmount monte el directorio compartido en el directorio llamada (si no esta creado, creelo) /NT/Compartidos (Compartidos es el nombre del recurso en Windows).

3.- Use comandos de navegación de sistemas de archivos para ver el recurso compartido ya en GNU/Linux. ¿Nota usted alguna diferencia entre la manera de listar de su ext3 y el NTFS?

RESUMEN

Los siguientes tópicos fueron discutidos en este capítulo:

- Samba es una implementación de servicios de archivo e impresión basados en el Protocolo SMB usados en redes NetBIOS sobre TCP/IP.
- Samba consiste en dos partes: daemons servidores y programas en el lado del cliente.
- El Servidor Samba se ejecuta como dos daemons:
  - El daemon nmbd que provee servicio de resolución de nombre y puede actuar como un servidor WINS en una red NT
  - El daemon smbd hace realmente la compartición de archivos e impresoras
- El servidor de Samba es configurado usando el archivo /etc/smb.conf
- El archivo /etc/smb.conf contiene una sección [global] y una sección para cada archivo o impresora que comparte. Este archivo es muy similar a los archivos .ini de Windows 3.x.
- Hay varios programas clientes de Samba disponibles:
  - El programa smbclient que puede ser usado para recibir información acerca de los servidores SMB y sus recursos com partidos y que puede ser usado muy similar a un ftp para transferir archivos.
  - El programa smbmount que maneja montar y desmontar recursos compartidos de SMB
  - Soporte del kernel para el sistema de archivos smbfs es un requisito para poder montar volumenes compartidos SMB.
- Samba redirecciona la operación de impresión (cliente y servidor) atraves del sistema local de impresión UNiX. Este puede ser de los disponible en los sistemas UNiX como son BSD, CUPS, SYSV, LPRng. El por defecto de Samba es BSD y en GNU/Linux es CUPS.

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Qué protocolos usa el suite de Samba?

2.- ¿ Cúal es el rol de nmbd en Samba?

3.- ¿ Puede el smbclient ser usado para conectarse a otra estación de trabajo GNU/Linux ejecutando Samba?

4.- Escriba el comando que listara todos los recursos compartidos de una estación de trabajo de nombre abiertos.

APENDICE A

Redes del Sistema Operativo GNU/LINUX

Capítulo 1

PREGUNTAS PRE-EXAMEN

1. Defina una red.
®.- Es un grupo de computadoreas y otros dispositivos conectados entre si para poder comunicarse e intercambiar recursos.

2. ¿Por qué usar una red?
®.- Utilizar una red reduce los costos de propiedad e incrementa la productividad de comparticion de datos, servicios y recursos.

3. Liste las cuatros capas del modelo TCP/IP, luego liste 2 protocolos usados por cada una ellas.
®.- *Aplicacion:* *FTP, Telnet, HTTP*
®.- *Transporte:* *TCP,UDP*
®.- *Internet:* *IP,ICMP,ARP*
®.- *Network Access:* *Ethernet, Token Ring, FDDI, serial, X.25, ATM*

Ejercicios 1-1: Ubique los Documentos de los RFC

En este ejercicio, se le pide que ubique en el Internet los sitios de donde puede descargar los RFC. Usted debe investigar más de un sitio web ya que no todos los portales actualizan sus RFC con regular frecuencia.

1. Desde el buscador www.google.com en un browser ubique tres sitios web que mantienen el listado de RFC. Liste los sitios por su URL como repuesta.
®.- http://www.rfc-editor.org/rfc.html
®.- http://www.faqs.org/rfcs
®.- http://www.cis.ohio-stat.edu/hypertext/information/rfc.html

2.- Elija un sitio RFC y use el browser para ubicar los RFC 1250, 2068, y 1441. Liste los títulos de cada uno de ellos. Determine si el RFC es la última versión. Si no lo es, identifique cual es el más reciente. Usted deberá usar un editor de RFC que permita peticiones, como el que se encuentra en el sitio web www.rfc-editor.org/rfc.html. Elija buscar/Search desde el enlace índice de RFC que le permite buscar por número, titulo y autor.

- RFC #	Titulo del RFC	RFC más actual
RFC1250		
RFC 2068		
RFC 1441		

3. Responda estas preguntas basada en la información en los RFC 2600 y 1700.

a. ¿Es OSPFv2 un protocolo estándar del Internet?

¿Es RIP-2 un estándar?
®.- OSPFv2 es un protocolo standard de internet; es STD 54/RFC 2328.
®.- RIP-2 tambien es un protocolo estandard de internet; es STD 56/RFC 2453.

b. ¿En que año se constituyó POP3 como un protocolo estándar del Internet?
¿Y el TCP? ¿Y el IP?
®.- POP3 se convirtio en un estandar de internet en mayo de 1996.
®.- TCP/IP son estandares desde septiembre 1 de 1981.

c. ¿Cúal es el STD para el protocolo de servicios NetBIOS?
®.- El Protocolo de servicio NetBIOS es STD 19.

d. ¿Cúal es el rango de los números de puertos más reconocidos (well-known)?
®.- Estos son manejados por la IANA y el rango va desde 0 a 1023.

e. ¿Para que es utilizado el puerto 110 ?
®.- Este puerto es usado para POP3

f. ¿Cúal es el rango de los números de puertos registrados?
®.- Los puertos registrados que no son administrados por IANA van del 1024 hasta el 65535. Estos puertos pueden ser utilizados por un usuario normal, programas o procesos.

g. ¿Para que es utilizado el puerto 533 ?
®.- Este puerto es utilizado para broadcasts de emergencia

PREGUNTAS POST-EXAMEN

1. ¿De cuantas capas consiste el modelo OSI? ¿Cual es la capa que más se relaciona con hardware?
®.- El modelo OSI consiste en 7 capas. La capa fisica

2. ¿Que son los RFCs?
®.- Los RFCs significan Request for Comments. Estos son documentos que contienen estandares TCP/IP con informacion sobre definiciones de protocolos.

3. Liste tres protocolos comunes del Internet y los puertos que generalmente usan.
®.- HTTP puerto 80
®.- POP puerto 110
®.- FTP puerto 21

CAPÍTULO 2

PREGUNTAS PRE-EXAMEN

1. ¿Cuál es la función del daemon inetd?
®.- inetd escucha multiples puertos de red e inicia otros demonios por cada peticion de red recibida.

2. ¿Cuáles son los tres tipos de topologías lógicas de redes?
®.- Bus, Anillo y estrella.

3. ¿Cuáles dos protocolos pueden ser usados para conectarse al Internet vía modem? ¿Cuál es mejor, y por qué?
®.- SLIP y PPP, PPP es mejor porque ademas de ser más reciente, provee autenticacion y puede encapsular multiples protocolos de red.

4. ¿Cuál es el rango de las tres principales clases de direcciones?
®.- 1.0.0.0 – 126.255.255.255
®.- 128.0.0.0 – 191.255.255.255
®.- 192.0.0.0 – 225.255.255.255

5. ¿Por qué existe la necesidad del Ipv6?
®.- Debido al crecimiento de internet, se sospecha que pronto se acabaran las direcciones ip, por lo tanto, ipv6 fue creado para eliminar este problema.

Ejercicio 2-1: Identifique la Dirección de Hardware en GNU/Linux

En este ejercicio, usted va a localizar la dirección del hardware de su computador en GNU/Linux. No se proveen soluciones a este ejercicio.

1.- Ingrese al sistema como root.

2.- El prompt digite:
®.- # ifconfig

3.- Aparecerá la información de su IP y la información de su NIC. Observe la dirección de su hardware (HWaddr).

Ejercicio 2-2: Ver el Cache del ARP en GNU/Linux

En este ejercicio, echaremos un vistazo al cache del ARP, entonces procederemos a agregar y eliminar entradas en el cache. No se proveen soluciones para este ejercicio.

1. En el prompt de root (#), escriba el siguiente comando:
®.- # arp

2. Visualice las entradas del cache de ARP. Si una entrada no existe de su segundo equipo, creala ejecutando este comando en el prompt:
®.- # ping [dirección IP de segundo sistema no existe]

3. Presione las teclas CONTROL+C para detener el proceso de ping. Visualice de nuevo el cache de ARP, escribiendo este comando:
®.- # arp

Ya debe existir una entrada para su segun sistema en el cache de ARP.

4. Escriba la entra ARP de su segundo sistema. Incluya solamente las direcciones IP y de hardware:

Dirección IP: ___________

Dirección de HW:_________

5. Si existen entradas adicionales en su cache de ARP, intente determinar a que nodos de la red pertenecen- Por ejemplo se ha usted comunicado recientemente con otro equipo en la red?

6. Para eliminar una entrada de ARP, use la opción -d. En el prompt simplemente escriba:
®.- # arp -d [Dirección IP segundo equipo en el sistema]

7. Visualice el cache del ARP escribiendo:
®.- # arp

Note como el ARP de su segundo sistema ya no aparece como entrada en el cache ARP.

Ejercicios 2-3: Convertir direcciones de Internet en valores decimales y binarios.

En este ejercicio, usted deberá convertir direcciones de Internet de binarias a decimales y vice versa. Escriba su respuesta en el espacio en blanco.

A.- Convertir de valores de binarios a decimal

1. 0111100000000100000011 11110000
®.- 120.1.3.240

2. 11000011 01010101 10011001 11010
®.- 195.85.153.210

B.- Convertir de valores de decimales a binarios

3. 207.199.32.205
®.- 1101111 11000111 00100000 11001101

3. 151.2.254.60
®.- 10010111 00000010 11111110 00111100

Ejercicio 2-4: Determinar las Clases y Direcciones IP Validas

En este ejercicio, usted deberá determinar la clase de cada dirección IP y además que sea una dirección IP válida para un computador. Si no lo es, explicar porque.

	Dirección IP	Clase	Valida? Si o No	Si no es valida, ¿por qué no?
1.	192.23.111.8	C	SI	
2.	10.1.1.256	A	NO	
3 .	148.108.62.95	B	SI	
4.	127.0.0.1	A	NO	Loopback
5.	245.255.123.49	E	NO	Experimental
6.	100.54.100.90	A	SI	
7.	162.34.0.0	B	NO	Es una subred
8.	127.65.18.191	B	NO	Reservada
9.	1.1.1.1	A	SI	
10.	208.152.84.255	C	NO	Broadcast subred
11.	225.37.257.34	D	NO	3er octeto sobre 255
12.	255.255.255.255	N/A	NO	Broadcast

Ejercicio 2-5: Direccionamiento de IP

En este ejercicio, vamos a ver las direcciones IP de la red de la Compañía X. No hay soluciones para este ejercicio.

Red de la Compañía X

El administrador de la red de X ahora necesita decidir en un nuevo esquema de direcciones IP para la redes nuevas de la compañía. Se le pide una propuesta para la compañía.

Las necedades expresadas por los departamentos incluyen:

- Trabajadores en la sede principal, realmente sólo necesitan acceso al sistema de la red interna de la Compañía X más la habilidad de navegar el Internet.
- El equipo de desarrollo necesitan acceso completo al Internet.
- El equipo de ingenieros esta preparado para considerar cualquier propuesta pero están preocupados acerca de costos de equipos y quien va a dar soporte a toda la infraestructura de la red.

Preguntas a considerar:

1.¿Debe X usar una red de clase A, B o C o múltiples redes de una clase en particular?

2.¿Cómo se asignaran números de redes a las redes en su diseño?

3-¿Debe X considerar establecer una intranet para ahorrarse direcciones IP?

4. ¿Qué otras opiniones son arrojas de estas consideraciones?

Ejercicio 2-6: Determinar las Mascaras de Sub-red por Defecto

En este ejercicio, debes determinar la mascara de sub-red por defecto de cada dirección IP.

1. 17.223.13.222

®.- 255.0.0.0

2. 194.10.99.2
®.- 255.255.255.0

3. 211.34.126.10
®.- 255.255.255.0

4. 152.4.202.69
®.- 255.255.0.0

5. 128.156.88.1
®.- 255.255.0.0

Ejercicio 2-7: ifconfig

Este ejercicio demuestra el uso del comando ifconfig, el cual es usado para ver y configurar las interfases de red. Debes ingresar como root para completar este ejercicio.

1. Mostrar la configuración actual.
®.- # /sbin/ifconfig

2. Mostrar la configuración actual de la primera tarjeta Ethernet.
®.- # ifconfig eth0

3. Escribe debajo tu dirección IP y mascara de red.
®.- 10.0.0.170   255.0.0.0

4. Desactive la interfaz ethernet primaria.
®.- # ifconfig eth0 down

5. Cambia tu dirección IP a 10.10.10.10, con una mascara de red de 255.255.0.0 y activa la interfaz.
®.- # ifconfig eth0 10.10.10.10 netmask 255.255.0.0 up

6, Muestra los ajustes en la interfaz para que sea segura y correcta.
®.- # ifconfig eth0

7. Adhiera un alias a la interfaz de 10.10.10.9 netmask 255.255.0.0.
®.- # ifconfig eth0:1 10.10.10.9 netmask 255.255.0.0 up

8. Visualice la tabla de enrutamiento.
®.- # route -n

9. Restaura su dirección IP original. Reinicia si tienes problemas para que su conexión a la red trabaje nuevamente.
®.- service network restart

Ejercicio 2-8: tcpdump

Este ejercicio da un repaso completo a través del uso del utilitario tcpdump, el cual permite ver los paquetes de la red y como viajan a través de la red. Para hacer este ejercicio, necesitarás iniciar como root y tener la tarjeta de red en modo promiscuo.

1. Muestra la configuración actual de su tarjeta de red:
   ®.- # ifconfig eth0

2. Ejecute el comando tcpdump sin opciones:
   ®.- # tcpdump

3. Observe la información de cada paquete. La mayoría usted no la entenderá, pero podrá descifrar algunas líneas. Si no existe tráfico en la red deberá generarlo desde otra computadora en la subred.

4. Cambiese a otra consola virtual o xterm. Muestra su tarjeta de red de nuevo:
   ®.- # ifconfig eth0

   Esta sentencia le notificará que la interfaz de red tiene la bandera PROMISC activada. Esto es porque tcpdump le dice a la tarjeta de red que quiere ver paquetes destinados para cualquier nodo en la sub-red, no sólo los paquetes direccionados a la dirección MAC de el sistema en el que usted esta.

5. Vuelva a la consola original. Detenga a tcpdump con CONTROL+C. Inicialo otra vez con el siguiente comando:
   ®.- # tcpdump -x -v

6. Muestre la salida de nuevo. Fijese que hay más información mostrada acerca de cada paquete. La opción ®.- x descarga la cabecera en su código hexadecimal y la opción -v le dice que muestre todo el progreso en pantalla imprimiendo más información acerca de cada paquete. Detenga el programa, después de haber visto varios paquetes.

7- El programa tcpdump viene con un lenguaje de filtrado que se puede usar para mostrar ciertos paquetes. Se puede filtrar por dirección de origen y destino, número de puerto, protocolo, tamaño del paquete y varios otros ajustes. Revise las páginas man para más detalles. Ejecute el comando mostrando solamente los paquetes ICMP:
   ®.- # if tcpdump -x icrap

8. Efectué un ping a una máquina desde otra máquina en la misma subred:
   ®.- # ping -c 1 sistemaB

Todo lo que deberá ver es una petición ICMP y una respuesta. El protocolo que el comando ping usa.

9. Use el siguiente comando ping:
   ®.- # ping -p abcd1234 -c 1 sistemaB

Fijese que usted puede ver el abcd1234 en los paquetes ICMP.

10. Usted puede filtrar exactamente paquetes yendo desde un sistema a otro:
    ®.- # tcpdump src 192.168.0.1 and dst 192.168.0.2

11. Usted puede hasta filtro solamente tráfico Web:
    ®.- # tcpdump src 192.168.0.1 and dst 192,168.0.2 and port 80

12. Estudie a ver si puede idear otros filtros interesantes.

13. Si estas ejecutando el X, trate de ejecutar el programa ethereal. El provee un buen GUI para el programa tcpdump. El utiliza la mismas sintaxis de filtrado pero provee más información en un formato que es un poco más fácil para leer.

Ejercicio 2-9: El IPv6 y el IPv4

Una aplicación ejecutándose en un cliente (A) implementando IPv6 envía data a un servidor (B) que implementa IPv4.

1.- Describa como la data avanza a través de la pila TCP/IP en ambos A y B y que dirección de destino es usada por B en cada etapa (no se preocupe por la dirección IP de A para este problema).

Ejercicio 2-10: Concerniente al IPv6

Es necesario proveer un reporte de estrategia de redes, y en particular un problema de asignación de dirección y como su COMPAÑÍA puede continuar proveyendo nuevas direcciones IP para los nuevos hosts durante la creciente expansión.

1.- La solución es obviamente migrar hacia el IPv6, ¿pero como?

Pruebe a ver si puede sugerir una estrategía que cubra los siguientes puntos:
A.- ¿En cuantas etapas por separado cambiaría usted por completo la red de la COMPAÑÍA?
B.- ¿En que orden cambiaría usted cada una de esta etapas?
C.- ¿Donde es lo más probable que necesitaría usted nuevo equipo/software?
D.- ¿Dejaría usted redes bajo el IPv4 en cualquier parte de la red de la COMPAÑÍA?
E.- ¿Cual otro beneficios derivados del IPv6 podría usted ofertar a los diferentes departamentos además de la no falta de direcciones IP?
F.- ¿Puede usted también tomar ventaja de las características del IPV6 para ofertar algunas mejoras potenciales en el futuro inmediato?

PREGUNTAS POST-EXAMEN

1. ¿Qué archivo controla la configuración del inetd?
®.- /etc/inetd.conf

2. ¿Cuáles son las tres clases de direcciones IP?
®.- Clase A, B y C

3. ¿Qué hace el Protocolo de Resolución de Direcciones(arpa)?
®.- El protocolo ARP resuelve direcciones ip a MAC. Este protocolo realiza un broadcast a todos los hosts en la red LAN si no los tiene listados en su tabla. Cuando recibe una respuesta de que una direccion MAC dada posee determinada IP, entonces la agrega a su tabla ARP para posterior uso y asi no necesita realizar broadcast de nuevo.

4. ¿Qué son los tres componentes para la conexión del PPP?
®.- Mecanismo de Encapsulacion, Protocolo de Control de Enlace y Protocolo de Control de Red.

CAPÍTULO 3

PREGUNTAS PRE-EXAMEN

1. ¿Qué es una netmask (mascara de red)?
®.- Una mascara de red es una cadena de puntos decimales que es utilizada para enmascarar una direccion IP y distinguirla entre los bits de red y los bits de host.

2. ¿Cómo es que funciona enrutamiento/routing?
®.- Un paquete IP, si no esta destinado a un host de su misma red, es enrutado a traves de un gateway(pasarela de red) a otra red. Como elegir la ruta depende de las entradas en la tabla de enrutamiento.

3. ¿Por que usamos enrutamiento Dinámico?
®.- Utilizamos la tabla de enrutamiento dinamico porque en grandes redes es extremadamente dificil mantener las tablas de enrutamiento dinamicas. En ese sentido, las tablas de enrutamiento son actualizadas por los protocolos de enrutamiento dinamico.

4. Dentro de una red Clase C, ¿Cómo podemos direccionar más de 255 máquinas?
®.- Esto es posible utilizando subnetting. esto ayuda a utilizar un conjunto existentes de direcciones IP más eficientemente creando algunos de los bits del host de la direccion ip en bits de red.

5. ¿Cuándo es que un gateway se convierte en un firewall?
®.- Solamente Cuando un paquete reenviado es deshabilitado.

Ejercicio 3-1: Subnetting/Subnetear

Tomemos las siguiente información en consideración antes de subnetear en ambiente de una compañía X. Esta compañía a decidido expandir y está en la necesidad de crear nuevos departamentos que necesitan ser interconectados a la red.

En la actualidad hay dos departamentos usando la siguiente configuración:

• Depto 1 usa 201.40.25.0 red de Clase C y tiene suficiente direcciones para hasta 20 hosts.

• Depto 2 usa201.40.26.0 red de Clase C que tiene suficiente direcciones para hasta 35 hosts.

Deseamos agregar dos Departamento más: uno que tiene 15 hosts y el otro con 35 hosts.

El único problema es que no tenemos más redes Clase C disponible para usarlas en estos nuevos departamentos. Tendremos que emplear una solución de subneteo para tomar las Clase C ya existentes y dividirlas para acomodar los nuevos hosts.

Además, debemos planificar cambiar para sólo usar una red Clase C única ya que en el futuro necesitaremos más redes para otros departamentos.

Deberá planificarse para:

A) El cambio al nuevo esquema de subredes en cuatro departmento usando dos redes Clase C.

B) El cambio a cuatro departamento con una red única Clase C.

1. Detalle cual departamento usará cual red, la mascara de la subnet a usar y cuantos hosts tenemos disponi-

bles y con que rango de direcciones IP. Tome en cuenta cualquier router que desearía usar entre los departamentos para llegar a sus conclusiones.

Ejercicio 3-2: Conceptos de Enrutamiento

1. Considere la siguientes entradas en la tabla de enrutamiento.

Route	**Netmask**	**Destination**
132.10.56.4	**255.255.255.255**	**142.12.201.2**
132.10.0.0	**255.255.0.0**	**142.12.201.1**
132.10.20.0	**255.255.255.255**	**132.10.4.18**
15.0.0.0	**255.0.0.0**	**132.1 0.4.1 B**
132.10.56.5	**255.255.255.255**	**132.10.56.4**
132.10.20.5	**255.255.255.255**	**142.12.201.2**
142.12.0.0	**255.255.0.0**	**Local interfase**
0.0.0.0	**0.0.0.0**	**142.12.201.2**

Donde (por ejemplo, a cual dirección IP) el enrutador entregará los paquetes direccionados a las siguientes destinaciones? (Use la tabla anterior para responder.)

15.127.243.8
132.10.56.4
132.10.20.3
132.10.20.5
132.10.200.3
37.92.129.1

2.- Considere el siguiente diagrama de una LAN.

Una máquina de prueba (A) es colocada en la LAN y se desea revisar su configuración IP. Se logra esto efectuando el comando ping a la máquina (B) en la misma red. El echo retorna; todo es bien entonces. Entonces intentamos hacerle ping ping a la máquina (C) que se encuentra en otra red. Esta vez el ping falla- ningunos son retornados. Verificamos que mi máquina tiene una ruta desde A a C via el router. ¿Entonces por qué el ping no funciona?

3. En la Compañía X, es hora de decidir en el enrutamiento entre el enrutamiento de la redes de X.

La tarea es dibujar las tablas de enrutamiento para los routers en el diseño de la red de su grupo. Si usted aún no ha asignado algunas clases de IP a la redes en su plan, deberá hacerlo ahora.

¿Percibe usted alguna área de problema /temática?

Ejercicio 3-3: Examinar las Tablas de Enrutamiento

Podemos examinar las tablas de enrutamiento del sistemas usando el comando netstat con la opciones: **-r o -e**

```
$ netstat -r          # route -e
```

A menudo es útil desplegar los hosts y las redes en la tabla en termino de sus direcciones IP y no sus nombres. Este es muy útil en particular cuando estamos usando servidores de nombre de dominio como es el DNS (lo cual se discutirá más adelante). Para efectuar esta acción, escriba el siguiente comando:

```
$ netstat -rn
```

Verá una entrada para la interfase de red y también una para la interfase loopback. En esta etapa, podrás ver la pequeña diferencia entre las dos salidas.

Pruebe las opciones -v y -ee del comando route; ¿qué información nos arroja cada una?

Sin una entrada en la tabla de enrutamiento que nos diga como llegar a una red o a un host, no nos podremos comunicar no con la red ni el host. Para ilustrar esto, escriba el siguiente comando:

$ ping Dirección_IP

Donde la Dirección_IP es la dirección de uno de los sistemas en otro segmento de la red. Note los errores de mensajes que se despliegan. Ahora ejecute este comando y también observe los mensajes de errores:

$ telnet Dirección_IP

Ejercicios 3-4: Protocolos de Enrutamiento Dinámico

En este ejercicio, trabajaremos con preguntas relacionadas con el tema de enrutamiento Dinámico para la Compañía X.

1. RIP usa un broadcasts cada 30 segundos para anunciar las rutas disponibles y el número de saltos a ellas. En esta ruta, dos enrutadores están sirviendo tres redes y ambos enrutadores están ejecutando RIP (versión 1). Complete los detalles de las rutas anunciadas y los saltos después que el sistema ha normalizado sus operaciones.

2. Ahora si el enlace de R2 a la Red A se cae. ¿Qué le pasaría a las rutas y los saltos anunciados por ambos R1 y R2 en la reded B y C? ¿Qué más puede pasar?

3. El administrador de Red de Compañía X esta considerando si es apropiado usar enrutamiento Dinámico. El administrador está enfrentando la siguiente situación.

La Compañía ha crecido de nuevo y ahora desea conectar otro sitio a su red corporativa. Para hacer esto, el administrador desea colocar otra conexión de 1-Mbps dentro de ambos sitios, así suplementando una conexión existente de 2-Mbps entre ellas. La red WAN debe verse entonces así: IMAGEN AQUI

La ventajas de este ordenamiento esta supuesto a ser resistente en el caso de fallas de uno de sus enlaces. Enrutamiento puede automáticamente cambiarse.

Usted debe tomar en consideración:

- ¿Vale la pena que la Compañía X considere ejecutar un protocolo de enrutamiento sobre los enlaces WAN?
- Y se es así, ¿cual protocolo(s) pueden ellos ejecutar?
- ¿Tiene la escogencia de un protocolo sobre otro algún beneficio?¿Cuáles son estos beneficios?
- ¿Existe algún beneficio en ejecutar enrutamiento Dinámico dentro de la Compañía X?

PREGUNTAS POST-EXAMEN

1. ¿En que se diferencian los conceptos de supernetting del de subnetting?

®.- El subnetting da más espacio de direcciones sin cambiar la clase de la red (incrementando el numero de ip disponibles). Por otro lado, supernetting lleva más enrutamiento reduciendo el tamano de las tablas de enrutamiento.

2. ¿Por qué es que no podemos usar direcciones que terminen en 0 o 255 como direcciones de una subnet?

®.- Las direcciones finalizadas en 0 y 255 son designadas para la ruta por defecto y las direcciones broadcast respectivamente.

3. Considerando RIP y OSPF ¿Cúal de estos protocolos de enrutamiento es más avanzado y por qué?

®.- RIP es un miembro de el grupo de protocolos vector-distancia. Por otro lado OSPF es un miembro del más avanzado grupo de protocolos de estado de enlace. OSPF no transmite rutas a traves de la red, pero en su lugar, actualiza el estado de los enlaces directamente conectados. Asi, OSPF provee trafico reducido y más seguridad.

4. ¿Después de cuantos saltos considerará RIP un destino no alcanzable?

®.- Despues de 16 saltos o más

CAPÍTULO 4

PREGUNTAS PRE-EXAMEN

1. ¿Qué significa IP?

®.- Internet Protocol (Protocolo de Internet)

2. ¿Es el encabezado de IP una fuente principal del sobre peso en la mayoría de implementaciones de IP?

®.- Si

3- ¿Cuáles tres campos son contenidos en el cabezal IP?

®.- Identificacion, Compesacion de fragmento, banderas de campo

4. ¿Cuáles números de puertos son reservados para servicios bien conocidos?

®.- 1024 hacia abajo

Ejercicio 4-1: Seguridad de la Red de la Compañía X

No se proveen soluciones a este ejercicio. La red está ya en un estado muy avanzado. Usted es el administrador de la red. Aqui se le provee las necesidades de diferentes departamentos de la Compañía X, y usted está en el deber de proveer estas necesidades pero siempre tomando en cuenta la necesidades de seguridad.

Es necesario que piense en todo el diseño de la Compañía X. Trate de responder las siguientes preguntas:

1.- ¿Cuáles medidas de seguridad de la red puede usted usar?

2.- La oficina principal no esta expuesta al Internet. ¿Qué tipos de defensas son apropiadas para estos sistemas?

3.- El departamento de Ingeniería necesita la máxima protección. ¿Qué sugiere usted aquí en este caso?

4.- El equipo de desarrollo necesita acceso rápido y fácil a recursos en línea pero posee datos sensibles que no pueden ser robados. ¿Puede usted pensar en una solución que satisfaga ambas de estas necesidades?

5.- También considere que otras sugerencias y medidas de seguridad puede usted tomar para asegurar que nuestra oficina no sea blanco de ataque.

Ejercicio 4-2: Completar la Configuración de Red

Este es un ejemplo del archivo hosts:

```
# Mi archivo hosts
127.0.0.1        localhost
192.168.1.5      almacen
192.168.2.7      proxy
```

Ahora usted debe poder hacer ping a estos equipos solo usando el nombre y no el IP. Aseguro que usted puede hacer esto.

Pruebe haciendo ping a un sistema en otra red (use la dirección IP del sistema). ¿Qué mensaje usted ve?

Ahora asignele a uno o más de los hosts en su archivo hosts un alias. Revise que ahora pueda contactar el sistema usando:

- Dirección IP
- Nombre de Host
- Alias

PREGUNTAS POST-EXAMEN

1. ¿Qué ventajas tiene el TCP sobre el UDP? ¿Cual es usado más a menudo?

®.- TCP es más flexible y más confiable en correccion de errores. TCP es más utilizado.

2. Describa el Cabezal IP.

®.- El cabezal IP contiene 3 campos para ayudar con el proceso de fragmentacion. Estos son Identificacion, compenzacion de fragmentos y banderas de campo.

CAPÍTULO 5

PREGUNTAS PRE-EXAMEN

1. ¿Cuáles son los tres pasos dados para comprobar para saber si hay conectividad?

®.- Verifique su cable de conexion, Verifique los paquetes de ping perdidos; una conexion lenta podria ser la causa de la perdida. Verifique si su conexion tambien trabaja o si sus drivers de red estan cargados.

2. ¿Durante el uso de un ftp, como podía corrumperse un archivo?

®.- Si el modo de transferencia no es el correcto.

3. ¿Qué efectua el netstat?

®.- Monitorea estadisticas de la red en el sistema local, puertos, estatus de TCP o UDP, tablas de enrutamiento, o usode recursos del sistema.

4. ¿Cuáles son los tres componentes de un sistema moderno de la dirección de la red?

®.- Objetosadministrados, administrador y agente.

Ejercicio 5-1: Instale y Configure una tarjeta de Ethernet Como Interfaz de Red

En este ejercicio vamos a instalar correctamente una tarjeta Ethernet como interfaz de red. En este ejercicio, la tarjeta Ethernet es una 3COM 3C905.

1. Ingrese al sistema como root.
2. Para buscar el módulo para utilizar para el dispositivo de red 3COM 3C905, escriba el siguiente comando para buscar el módulo apropiado:
   /sbin/modprobe -l

   Para enumerar los módulos una página a la vez, escriba:
   /sbin/modprobe -l | more
3. El módulo en este caso será 3c59x.o. Escriba lo siguiente para instalar el módulo:
   /sbin/insmod 3c59x
4. Para ver si el módulo ha sido instalado correctamente, digite:
   /sbin/lsmod
5. Después de asegurarse que el módulo está presente, utilize el comando ifconfig para configurar la tarjeta de Ethernet. Asegúrese de que la tarjeta Ethernet está alias al interfaz correcto (eth0, eth1, eth2, etc.) en /etc/conf.modules en el archivo, sólo debe escribir:
   # cat /etc/conf.modules

   Si el interfaz no es el deseado, utilice un editor de texto, como pico y modifiqúelo. Ahora utilice el ifconfig para asignarle una Dirección IP:
   /sbin/ifconfig eth0 192.168.2.15 up

   Esta sentencia asigna la IP ADDRESS 192.168.2.7 a la interfaz eth0 y la activa (up).
6. Ahora es necesario manipular la tabla de enrutamiento para establecer las rutas estáticas en la red:
   /sbin/route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0

   Esta sentencia establece una ruta a la red 192.168.2.x atraves de la interfaz eth0
7. Por último utilice el comando ifconfig para ver si la tarjeta Ethernet ahora se está presentando activa en la red:
   # /sbin/ifconfig

Ejercicio 5-2: Implementación del Nuevo Producto de Administración de Red

Su tarea como administrador es planificar la implementación del nuevo producto en toda la sede principal de su empresa. Le será necesario tomar en consideración los siguientes puntos:

1.- ¿Cuáles parámetros del sistema y de la red dene usted medir?

2.- Cuáles son los nodos críticos en la red existente en la compañía en este momento?

3.- ¿Dónde tiene sentido colocar la estación de trabajo que efectuará la administración?

4.- ¿Existen algunas situaciones de red comunes que usted puede programar cierto tipo de respuesta automatizada?

PREGUNTAS

1.- ¿Cómo puede usted comprobar si cualquier parte del TCP/IP ha registrado un error?

2.- ¿Cómo puede usted probar su espacio de nombre (namespace) de dominio?

3.- ¿Cuáles son algunas herramientas para la administración de una la red moderna?

CAPÍTULO 6

PREGUNTAS PRE-EXAMEN

1. De dos razones del porqué el servicio de nombres no es efectuado en una forma centralizada.

®.- Los servicios de nombres son distribuidos por 3 razones :
- Eficiencia: Los nombres locales son resueltos localmente.
- Confiabilidad: Si hay una falla en el servidor central no se detendra todo el sistema.
- Flexibilidad: Los nombres locales no son registrados centralmente.

2. ¿Cúal es el nombre del directorio raíz del árbol del DNS?

®.- La raiz del arbol DNS es llamado “” (NULL)

3. ¿Qué es un FQDN (Fully Qualified Domain Name)?

®.- Es el nombre del host junto con su nombre completo de dominio.

4. ¿Qué es un resolver?

®.- El resolver es la parte de su software TCP/IP que sabe como contactar los servidores de nombres y realiza peticiones de DNS

5. ¿Son las siguientes sentencias falsas o verdaderas?

- El DNS es independiente de la topología de la red.

  ®.- Verdadero

- Un resolver siempre sabe la localización del servidor de nombre ráiz/root.

  ®.- Falso

- El resolver efectúa un cache de la data.

®.- Falso, el servidor lo hace.

Ejercicio 6-1: Configurar un Servidor DNS

Escriba los archivos de configuración del servidor de nombre en un subdirectorio de su home. Necesitará el la ruta completa a este directorio para colocarla luego en lugar de (nombre_de_Directorio).

1. Escriba el archivo de arranque/boot, named.conf, para su dominio:

```
# BIND v8 named.conf para el servidor primario de la zona  Dominio
# opciones del servidor
options {
            directory   "nombre_de_Directorio";
};
# zonas
zone  "."   in  {
            type  hint;
            file "root.cache"
};
zone  "0.0.127.in-addr.arpa"  in  {
            type master;
            file "db.127.0.0";
};
zone  "Dominio"  in  {
            type master;
            file  "db.Dominio";
};
```

2. Escriba el significado de cada línea menos las secciones que son comentarios.

Ejercicio 6-2: Archivo de cache del Servidor Raíz/Root

A los archivos de data le podemos dar cualquier nombre, siempre y cuando el archivo de configuración named.conf se refiere a el por ese nombre.

1. Cree el Servidor Raíz/Root Server, el archivo se deberá llamará root.cache:

```
;   root.cache
;  pre-cargar el servidor primario con la información
;   a cerca de los sservidores root
                    99999999      IN      NSRootSrv.
RootSrv.            99999999      IN      A       RootIP
```

Tome en cuenta que el RootSrv debe ser el FQDN del servidor root (como es ns.abiertos.gov.).

2. ¿Por qué necesitamos un record A (Address) para el servidor root?

Ejercicio 6-3: Archivo Data de los Hosts (Opcional)

Escriba el archivo data de los hosts para su subdominio, web.Dominio.org:

```
;   información del host para la zona Dominio.
;
$TTL         86400
Dominio.     IN    SQA SrvPrimario.     postmaster.SrvPrimario.
                                        1;   Serial
                                        10800;   Refrescar después de 3  horas
                                        3600;     Reintento después de 1  hora
                                        604300;   Expira después de 1 semana
                                        86400;  TTL Minimo de 1 día
Dominio.                               IN  MS  SrvPrimario.
SrvPrimario                            IN  A    IPPrimario
;  información de los hosts individuales...
host1.Dominio.                         IN A IP_host
```

Si se le dificulta entenderlo, hechemos un vistazo al siguiente ejemplo:

```
; información del host para la zona xyz.edu
$TTL                    86400
xyz.edu.                IN  SOA     escuela.xyz.edu.        postmaster.xyz.edu.   (
                                                1;          Serial
                                                10800;   Refrescar después de 3 horas
                                                3600:     Reintentar después de 1 hora
                                                604800: Expira después de 1 semana
                                                86400):  Minimo TTL de 1 día

xyz.edu.                IN        NS     escuela.xyz.edu.
escuela.xyy.edu.        IN        A            200.42.200.43
; Información de los hosts individuales
pop.xyz.edu.            IN        A            200.42.200.47
matematica.xyz.edu.     IN        A            200.42.200.48
fisica.xyz.edu.         IN        A            200.42.200.49
quimica.xyz.edu.        IN        A            200.42.200.50
```

El record SOA (Start of Authority) indica que el host escuela.xyz.edu tiene autoridad sobre la zona xyz.edu. Es un record de recurso único, que se define sobre unas cuantas líneas usando parentesis. También específica la dirección de correo del administrador responsable de la zona (postmaster@xyz.edu), con la @ reemplazada con un punto.

El record NS lista los servidores para la zona xyz.edu; estamos usando solamente un servidor. El record A es un record glue. Sin este record los servidores de nombre supieran el nombre del servidor autorizado, pero no su dirección, y por esto no pudiesen comunicarse con el.

Finalmente, hay records del tipo Address para cada host en la zona; esta es la información que estamos realmente interesados y es la razón principal de la existencia del DNS.

Ejercicio 6-4: Archivo del Host Local

1. Escriba el archivo del host local, backup.127.0.0.

   El archivo del host local mapea la dirección de la interface loopback local (la cual normalmente es 127.0.0.1) al nombre "localhost". Recuerde que el record SOA (incluyendo el postmaster) todos deben estar en una misma línea:

```
$TTL        86400
0.0.127.IN-ADDR.ARPA.       IN  SOA   SrvPrimario.   postmaster.SrvPrimario.    (
                          1;  Serial
                          10800;   Refresca después de 3 horas
                          3600;  Reintenta después de 1 hora
                          604800;  Expira después de 1 semana
                          86400;   Minimo TTL de 1 día
                          )

0.0.127.IN-ADDR.ARPA.     IN   NS     SrvPrimario.
SrvPrimario.              IN   A       IPPrimario

1.0.0.127.IN-ADDR.ARPA.   IN   PTR     localhost.
```

2. Iniciar el servidor de nombre:

```
# cd   /etc/rc.d/init.d
# named start
```

   Revise el archivo log (/var/log/messages) para ver si el servidor de nombre si inicio correctamente o si hubiesen errores en sus archivos de configuración.

PREGUNTAS POST-EXAMEN

1. ¿Son las siguientes acertaciones ciertas o verdaderas?

   • Cada host debe ejecutar un servidor de nombre individual

   ®.- Falso

   • Un servidor de nombre puede ser primario a dos zonas.

   ®.- Verdadero

   • Un host debe saber la localidad de un servidor de nombre de su zona padre

   ®.- Falso, solamente uno para su propia zona

   • Un servidor de nombre debe servir por lo menos a una zona.

   ®.- Verdadero

2. ¿Por qué es que alguna data no es autorizada (data nonauthoritative)?

®.- La data nonauthoritative viene desde el cache de servidor de nombres más bien siendo traido desde un servidor de nombres.

3- Nombre un servicio de nombre alternativo.

®.- Servicios NIS, NIS+ y WINS

4. Nombre tres tipos de records de recursos.

®.- Un (IPV4), AAAA(IPV6), Nameserver, SOA

5. Hay dos convenciones usadas al elegir un dominio. ¿Cuáles son estas dos convenciones? Nombre dos ejemplos de cada una.

®.- Organizacional, tales como com, edu, gov, etc y ubicacion geografica como .do .fr , etc

CAPÍTULO 7

PREGUNTAS PRE-EXAMEN

1. Nombre tres clientes FTP.

®.- ftp, mozilla, ncftp.

2. ¿Qué significan las siglas LDAP y que hace este?

®.- Lighweight Directory Access Protocol. Este funciona como base de datos de informacion concerniente a cuentas de usuarios y recursos de red. Tambien puede ser utilizado para autenticacion de usuarios.

3. ¿Cuál servicio de red provee direcciones IP?

®.- Dynamic Host Configuration Protocol (DHCP)

4. ¿Cuál es la diferencia entre NTP, ntpd, y xntpd?

®.- NTP es el nombre del protocolo. Las implementaciones del protocolo NTP para unix conteniendo un demonio son llamados ntpd. xntpd es el nombre del paquete que es considerado actualmente la implementacion standard de referencia.

Ejercicio 7-1: Configurar Servicios en xinetd

Con el lanzamiento de la versión 7 Red Hat GNU/Linux, xinetd fué incluído y presentado como un reemplazo para inetd. Mientras que la configuración de ambos presenta ciertas diferencias entre xinetd y inetd, ambos son lo relativamente intuitivos para que nos podamos facilmente adaptar entre el formato de uno y el otro. En este ejercicio, demostraremos como poner en práctica cambios a un par de servicios dentro de la nueva estructura.

La soluciones a este ejercicio se incluyen en su contenido.

1. Arranque su sistema en Linux.

2. Ingrese como root.

3. Verifique que el ftp se puede ejecutarse para el localhost. Una vez que usted ve que la conexión no falló, presione CONTROL+C para salir del shell y regresar al prompt.

```
# ftp localhost
Connected to localhost.
220 localhost FTP server (Version wu-2.6.1(l) Ued Aug 9 05:54:50 EOT 2000)
ready.
530 Please login with USER and PASS.
Name (localhost:root): #
```

4. Ahora para deshabilitar el sendee, muévase al directorio del árbol de configuración del xinetd:

```
# cd /etc/xinetd.d/
```

5. Aquí usted encontrará los archivos que son usados para controlar los servicios. Use un editor de texto para abrir el archivo de configuración de ftp:

```
# vi wu-ftpd
```

6. Agregue las siguientes directivas de configuración:

```
disable = yes
```

Debe ser dentro de los corchetes. Un lugar bueno para ello sería después de la directiva.

```
r* 1 r*i
```

7. Guarde y cierre el archivo.

8. Reiniciar el super servidor xinetd:

```
# /etc/init.d/xinetd restart
```

9. Intentar conectarse por ftp al localhost otra vez. El servicio ahora debe estar cerrado.

```
# ftp localhost
ftp: connect: Connection refused ftp> 10.
Salga del ftp
```

Ejercicios 7-2: Configurar el DHCP

En este ejercicio se rquieren dos computadoras conectadas en red. Instalaremos un servidor DHCP en una de ellas. Necesitaremos un cliente en la otra para hacer la operación de prueba. En este ejercicio deberá usar la dirección de subred privada 192.168.0.0. No se proveen soluciones para este ejercicio.

1. Instale el dhcpd. En muchas distribuciones, ya este estará instalado. Si no es el caso, es recomendable que usted instale el paquete proveido por su vendor. Alternativamente puedes descargar el fuente desde el ftp.isc.org descomprimirlo, desempaquetarlo, compilarlo e nstalarlo.

2. Determine donde se encuentra su archivo dhcpd.leases. Puede que este en /etc o quizas en /var, como en /var/state/dhcp. Las páginas man de su paquete le puede ayudar a ubicar su archivo dhcpd.leases:

```
# man 5 dhcpd.leases
```

3. Si el archivo dhcpd.leases no existe deberá crearlo:

```
# touch /etc/dhcpd.lease
```

4. Edite el archivo /etc/dhcpd.conf para que contengan las siguientes líneas:

```
dcfault-lease-time 3600;
option  subnet-mask  255.255.255.0;
option  routers  192.168.0.1;
option domain-name-servers  192.168.0.1,   192.168.0.9;
option domain-name  "mi-dominio.org";

subnet  192.168.0.0  netmask  255.255.255.0 {
     range  192.168.1.20   192.168.1.250;
     option  netbios-name-servers   192.168.0.15;
}
```

5. Inicie el Servidor DHCP en modo de depuración (debug):
   # /usr/sbin/dhcpd -d -f

6. Inicie el cliente DHCP. En el sistema que debe estar configurado para usar DHCP y luego podemos revisar para ver que parámetros nos asigno.

7. Observe la salida de depuración que el daemon servidor dhcpd expide.

8. Revise los parámetros que el cliente DHCP recibió. En GNU/Linux, con en la mayoría de sistemas operativos podemos usar los comandos ifconfig y route sin argumento alguno.

9. Renove el alquiler de la dirección que recibió. Con el cliente de pump, usted puede efectuar esta tarea escribiendo la siguiente sentencia:
   # pump --renew

10. Verifique si puede renovar su alquiler de DHCP y recibir uno nuevo. Con el cliente pump, usted puede lograr esta tarea primero relegando el alquiler y luego contratar uno nuevo:
    # pump --release
    # pump --renew

    Con el dhcpcd, tendrá que matar el daemon, deshabilitar la interfaz y rehabilitarla.

Ejercicio 7-3: Configurar los Servicios de FTP Usando ProFTPD

Las soluciones a este ejercicio se incluyen en su contenido.

1. Instale el paquete de ProFTPD. Si su distribución viene ya con el incluido en una versión pre-empaquetada use esa. Sino descargue el paquete desde el sitio web de ProFTPD http://www.proftpd.net, descomprimale, compilelo e instalelo.

2. Debe crear un directorio para el acceso anónimo:
   # mkdir /home/ftp

3. Edite el archivo /etc/proftpd.conf. Primero configure el servidor para solo tener acceso de usuarios autorizados:

```
ServerName          "FTP de CodigoLibre"
ServerType          inetd
DefaultServer       on
Port                21
MaxInstances        30
User                nobody
Group               nobody
Umask               022
```

```
<Directory  /*>
    AllowOverwrite          no
</Directory>
```

4. Edite el archivo /etc/inetd.conf. Elimine el simbolo de comentario de la línea que pertiene a cucalquier servicio de FTP actualmente. Agregue la siguiente línea:

```
ftp    stream    tcp    nowait    root    /usr/sbin/tcpd    in.proftd
```

Usamos wrappers TCP aquí, wlo cual no es estrictamente necesario ya que ProFTPD puede efectuar control de acceso a los hosts por si mismo, pero ers bueno ser consitente y usar wrappers TCP en todos los servicios.

5. Le enviamos una señal al inetd para que lea nuevamente su archivo de configuración:

```
# klllall   -HUP  inetd
```

6. Pruebe el servidor ejecutando un cliente FTP. Es mejor provar desde un equipo diferente, siempre y cuando sea posible. Será necesario suplir un nombre de usuario y contraseña válidos en el sistema.
7. Habilite el acceso anónimo agregando las siguientes líneas al final de su archivo /etc/proftpd.conf:

```
<Anonymous   /home/ftp>
User                    ftp
Group                   ftp
UserAlias               anonymous ftp
MaxClients              10
  <Limit  WRITE>
     DenyAll
  </Limit>
</Anonymous>
```

8. Pruebe el servidor ejecutando un cliente FTP, desde un computador diferente si es posible. Ingrese (login) con el nombre de usuario “ftp”. Se le pedirá en el prompt entrar su dirección de correo en vez de una contraseña.

Ejercicio 7-4: Instalar y Compilar Squid

Para esta práctica necesita tener instalado un versión del compilador gcc y un ambiente de build válido. Además deberá estar ejecutan X, con un Mozilla ya instalado, para poder provar el squid al menos que sepa navegar en lynx. Preparecese para describir que va sucediendo. ¿Dónde se instaló el software? ¿Dónde están los binarios del squid? ¿Dónde esta el archivo squid.conf? Elimine cualquier copia adicional de squid.conf y debe crear un viínculo simbólico a /etc/squid/squid.conf en su lugar. Note que muchas distribuciones incluyen a Squid en sus instalaciones. Usted puede revisar si la suya es una usando el comando ls -R /etc | grep squid para ver si Squid ya está instalado. Un resultado similar al siguiente indicaría que si ya esta instalado:

```
$ Is   -R   /etc | grep squid
squid/
squid
squid*
/etc/squid:
squid.conf
```

Fijese que los resultados incluye los directorios pero no necesariamente las localidades. La línea squid* es el script de ejecusión, que se encuentra en /etc/rc.d/init.d/ para esa distribución en particular.

Puede ser que elija no efectuar los pasos del 1-6 si usted ya tiene instalada una versión de Squid, aunque algunas cosas en específico puede que cambie dependiendo de su distro. Esta información es lo más pro-

bable bueno para la mayoría de situaciones pero puede ser muy importante si usted no tiene un ambiente vealido de gcc. Las soluciones a este ejercicio se encuentran en el Apéndice A.

1. Dirijase al Internet y descargue la última versión estable del Squid.

- Verifique en la pagina www.squid-cache.org y utilice el cliente ftp de su gusto para obtener el codigo fuente mas reciente. Si cualquiera de estos pasos falla, podria necesitar modificar los permisos o simplemente ejecutar como root :

ncftpget ftp://uiarchuve.uiuc.edu/pub/www/squid/squid(version)/STABLE/squid(version)

2. Descargue el fuente y descomprimalo en el directorio apropiado.

- Extrraiga el codigo fuente desde el archivo tar a /usr/local/src:

tar -xvzf squid(version).tar.gz -C /usr/local/src

3. Siempre lea los archivos de instrucción INSTALL, Readme y QUICKSTART.

- cd squid(version)

- cat INSTALL

- less QUICKSTART

4. Proceda a configurarlo.

- ./configure

- Esta es la parte que requiere de un entorno gcc valido. para una completa instalacion de muchas distribuciones e instalaciones de clases, esto debe ser incluido.

5. Ejecute el make all para compilar el software.

- make all

6. Ejecute la sentencia make install para instalar el software. Deberá ser root para instalar el softeare por cuestión de permisos.

- make install

7. Edite el archivo squid.conf para modificar los valores apropiados para las siguientes variables. Puede ser que solo tenga que modificar uno o dos de estas variables.

- El archivo de configuracion por defecto de squid podria requerir solamente un simple cambio. Utilice Vi o Emacs o su editor de texto preferido para editar el archivo squid.conf:

http_access allow all

Note que debe tener un servidor DNS para que squid trabaje. Puede especificar un servidor DNS valido con la variable dns_nameservers en el archivo squid.conf si estas presentado problemas con tu DNS local.

Cambie el parámetro http_access de denegar a aceptar (deny a allow). Usted podrá luego refinar estos parámetros para restringir acceso a redes, hosts y usuarios en específico.

http_access allow all

Puede que tenga que modificar para especificar un servidor DNS si la función ya no se encuentra ejecutándose en su sistema. Deberá quitar el comentario a la línea "dns_nameservers" y reemplazar "none" con la dirección IP apropiada:

dns nameservers 192.168.2.1

Si desea hacer que la notificación por e-mail funcione deberá especificar el correo electrónico del administrador. Elimine el comentario de la línea "cache_mgr" y reemplace Webmaster con la dirección de correo apropiada:

cache_mgr admin@codigolibre.org

8. De inicio al squid con el parámetro -z para crear la estructura de directorio de cache. Esto puede llevar a cabo por varios minutos una actividad masiva en sus discos.

9. De inicio al squid una segunda vez, ahora sin la opción -z, y verífique que el proceso eestá ejecutándose. Recuerde que squid requiere acceso al DNS (o localmente o un host en otra red) para operar.

10. Configure su navegador para que apunte al Squid.

11. Verifique que su navegador estasu navegador está funcionando a traves del Squid accesando una página Web. Usted puede verificar que lapágina está pasando por el Squid, usando el comando tail para leer el archivo /var/logs/access.log. Si no le registra, entonces verifique su archivo squid.conf, en particular los parámetros en el http_access.

Ejercicio 7-5: Sincronización del Tiempo de la Red

En este ejercicio, usaremos NTP como un cliente y como un servidor. Nesecitaras un sistema conectado al Internet sin un firewall de por medio para realizar la mayor parte de este ejercicio. Si no puedes tener acceso a un sistema con acceso directo al Internet, puedes saltar al paso 3 y cambiar la línea "servidor" en el paso 4 a las siguientes dos líneas:

server 127.127.1.0
fudge 127.127.1.0 stratum 10 refid time

Las soluciones a este ejercicio están incluidas en el contenido.

1. Instale el paquete NTP. Si su distribución viene con una versión pre-enpaquetada, usala. De lo contrario, deberás descargar el fuente de ntp.org, desempaquetarlo, compilarlo e instalarlo. Note que el paquete puede ser llamado por varios nombres. En Red Hat, es llamado xntp3, pero puede ser llamado ntp, ntpd, xntp o xntpd.

2. Si ya tienes un archivo /etc/ntp.conf, hasle un copia como copia de resguardo y eliminalo.

3. Ejecuta ntpdate a un servidor de tiempo público. Podemos sugerir como servidores de tiempo a ntp.css.gov, time.apple.com o time.mit.edu. Una lista completa de servidores de tiempo esta disponible en http://www.eecis.udel.edu/~mills/ntp/servers.htm.

   # /usr/sbin/ntpdate ntp.css.gov

4. Configure ntpd ahora como un cliente usando el mismo servidor de tiempo público. Debes crear el archi-

vo /etc/ntp.conf que contengan sólo las siguientes líneas:

```
server  ntp.css.gov
driftfile /etc/ntp.drift
```

Usted puede especificar múltiples líneas de servidores si le gustaría cuestionar múltiples servidores. Esto provee redundancia si un servidor esta fuera y aprovecha mejores niveles de exactitud.

5- Inicie el demonio ntpd. (Recuerde que algunas distribuciones llaman al ejecutable xntpd.)

```
#   /usr/sbin/ntpd
```

6. Espere unos cuantos minutos para permitir al demonio sincronizarse. Usa el programa xntpdc para verificar que todo esta corriendo apropiadamente. En el prompt xntpdc, usa el comando peers:

```
xntpdc > peers
remote          local           st   poll   reach   delay       offset      disp
ntp.css.gov     192.168.0.1     2    64     1       0.04797     0.00712     15.8750
```

Si “st” es 16, entonces no estas sincronizado con el servidor de tiempo. Espera unos minutos o intenta con otro servidor.

Corra ntpdate en otro sistema para asegurar que puedes tomar el tiempo y ajustarlo exactamente desde el servidor NTP.

Ejercicio 7-6: Configurar y Usar OpenLDAP

Las soluciones para este ejercicio están incluídas dentro del contenido.

1. Instale el paquete OpenLDAP. Si su distribución viene con una versión preempaquetada, úsela. Si este no es el caso tendrás que descargar el paquete con los fuentes, desempaquetarlo, compilarlo e instalarlo.

2. Debe crear un directorio /var/ldap para almacenar toda la información de directorio de LDAP:

```
#    mkdir   /var/ldap
```

3. Edite el archivo de configuración por defecto slapd.conf, haciendo los siguientes cambios a la sección que empieza con “database ldbm”. El archivo de configuración por lo general estará en el directorio /etc/openldap, pero la ruta puede variar dependiendo en como ha instalado el paquete.

```
suffix  “dc=codigolibre,   dc=org”
directory  /var/ldap
rootdn  “cn=admin,  dc=codigolibre,  dc=org”
rootpw mypassword
```

Normalmente, usarias una clave encriptada, pero para este ejemplo usaremos una clave de texto plano.

4. Kill/Mata cualquier demonio slapd ejecutándose:

```
killall   -TERM ilapd
```

5. Inicia el demonio slapd:

```
/usr/sbin/slapd
```

6. Edita el archivo /etc/ldap.conf, asegurándose que las siguientes dos entradas que permiten a las utilidades cliente LDAP conectarse a nuestro servidor LDAP:

```
host   127.0.0.1
base dc=codigolibre,dc=org
```

7. Debe crear un archivo LDIF para popular la base de datos. Llámelo ejemplo.ldif y editelo para que su contenido sea lo siguiente:

dn: dc=codigolibre, dc=org
objectclass: domain
dc: codigolibre

dn: cn=Miguel, dc=codigolibre, dc=org
objectclass: person
en: Miguel
sn: Usuario
mail : miguel@codigolibre.org

8. Use ldapadd para popular el directorio con el archivo LDIF que creo:
   $ ldapadd -f ejemplo.ldif -D "cn=admin, dc=codigolibre, dc=org" -W

   se le pedirá digitar la clave de root:
   Entra la clave LDAP :

   Escriba la clave que usted definió en el archivo de configuración mypassword.

9. Ahora que el directorio esta populado; ejecutaremos algunas busquedas en el. Note que no hay nada particularmente especial acerca del comando ldapadd que fué usado para popular el directorio; siempre y cuando usted tiene los derechos de acceso apropiados, usted puede agregar o modificar las entradas en cualquier momento usando los comandos ldapadd o ldapmodify. Las configuraciones por defecto son ajustadas con acceso de lectura para todos y lectura/escritura para el rootn que especifico en el archivo de configuración.

10. Busque una entrada con el nombre común de Miguel:
    $ ldapsearch -b 'dc=codigolibre,dc=org' 'CN=miguel'
    cn=Craig, dc=codigolibre, dc=org
    objectclass=person
    cn=Miguel
    sn=Usuario
    mail= miguel@codigolibre.org

    Usamos la opción -b para especificar donde iniciar la búsqueda. Normalmente, el por defecto es tomado desde el archivo ldap.conf. Fijese que la comparativa CN no es caso sensitiva.

11. Busque todos los usuarios que su dirección de correo electrónico contengan el nombre Miguel:
    $ ldapsearch 'mail=*miguel*'

    La salida sería la misma que la del paso anterior. Note el uso de metacaracteres (wildcards).

12. Imprima la salida de la base de datos completa a la salida estandar (pantalla):
    $ ldapseach -L "objectclass=*"

    Aqui imprimimos el contenido de todos los objetos en el directorio. La salida es un formato LDIF, asi les que se debe parecer exactamente al archivo de entrada usado para crear la base de datos. Asi que, puede usar la salida de este comando para modificar la información y reimportarla.

PREGUNTAS POST-EXAMEN

Las espuestas a estas preguntas están en el Apéndice A.

1. ¿Por qué DHCP normalmente no es usado para asignar direcciones a servidores?

2. ¿Cuál comando FTP deben los servidores FTP implementar para permitir acceso de clientes que están detras de un firewall?

3- ¿Como se inicia un servidor NIS maestro?

4. ¿Usted telnet en su cuenta ISP vía Internet. ¿Qué puede usted hacer para asegurarse que no le roben su contraseña?

5. De un ejemplo usando ldapsearch para localizar un objeto en una base de datos LDAP con el nombre común de Jose Paredes.

6. ¿Cuál protocolo usan los servidores Squid para comunicarse entre si?

7. ¿Cuál protocolo es usado para trasportar artículos de noticias Usenet?

CAPÍTULO 8

PREGUNTA PRE-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Quienes conforman el Apache Group?

2. ¿Cuál es el comando para instalar Apache desde RPMs?

3. ¿Dónde podemos obtener la última version de Apache?

4. ¿Cuándo inicia Apache, cómo se comporta? ¿Puede manejar múltiples peticiones de clientes a la vez?

5. ¿Qué puerto usualmente escucha Apache para aceptar peticiones?

Ejercicio 8-1: Instalación de Apache (Opcional)

El propósito de este ejercicio es utilizar los comandos básicos de GNU/Linux para instalar la versión de apache en la distribución de GNU/Linux que tengas. Este ejercicio asume que tienes una distribución basada en RPM. La solución a este ejercicio son suministrados en el Apéndice A al final de este manual.

1. Remueva algún paquete rpm de apache si esta instalado.

2. Instala el rpm de apache que vino con su distribución (o alguno actualizado descargado desde algun sitio de Internet).

3. Localiza los archivos de configuración de apache. Cuales son sus nombres ? Examina a traves de cada archivo para mirar su contenido.

4. Inicia el servidor Apache. (El programa de instalación ya ha iniciado el servidor por ti).

5. Utilice un explorador para asegurarse que el servidor esta trabajando.

6. Crea un cambio menor en la pagina inicial HTML que apache visualiza asi sabras si realmente es tu pagina. Examina la pagina de nuevo para asegurarte que el cambio ha pasado .

7. Utilice el comando ps para mirar que procesos están corriendo, incluyendo httpd

8. Para el servidor Apache.

Ejercicio 8-2: Configuración de Apache

En este ejercicio, haremos cambios en tres archivos clave de configuración de apache: httpd.conf, srm.conf y access.conf. La solución a este ejercicio son proporcionados en Apéndice A.

1. Crea una copia a los archivos de configuración

2. Invierte algun tiempo revizando el contenido de estos archivos para que te familiarices con las diferentes directivas en cada archivo de configuración.

3. Deten el servidor apache. Cambia el usuario bajo el cual corre httpd, desde el default, reinicia apache. Puedes verificar que el cambio fué satisfactorio utilizando ps.

4. Proporciona a su servidor un nombre conveniente.

5. Cambia la directiva ServerRoot. Asegúrese que crea copias de todos los archivos de configuración necesarios.

6. Ahora Cambie también la directiva ErrorLog.

7. Crea un directorio nuevo con un index.html diferente y otro conteniendo el que tu quisieras crear. Esto es bueno para practicar HTML. Ahora cambia el DocumentRoot del servidor. Utilice su explorador para verificar los cambios hechos. No olvide refrescar la pagina.

8. Asegúrese que posee diferentes cuentas de usuario en su sistema linux. Cree las páginas personales para estos usuarios y asegúrese que pueden explorarlas.

Ejercicio 8-3: Crear un Certificado con OpenSSL

Este ejercicio ilustrará el proceso de la creación del certificado contorneado previamente y permitirá que usted considere lo que parecería el certificado una vez en uso en un servidor web. No hay soluciones proporcionadas para este ejercicio..

1. Ingresa como root.

2. Crear un directorio temporal asi tendrá un lugar para trabajar con los archivos SSL.
   mkdir /ssltemp

3. Invoca el comando openssl con la función req para crear una certificado de petición.
   openssl req -new -out cert.csr - keyout key.pern

   Ingrese y verifique una clave cuando le sea solicitada. Memoricela ya que la necesitara para el resto del ejercicio.

4. Después de ingresar y verificar su clave, te preguntará por la información que esta colocada en el certificado. Sientase libre a ser creativo con estos campos. Cuando es solicitado para introducir una clave, solo presione ENTER. Si deseas, puedes introducir un nombre de compania.

5. Invoca openssl con la función rsa.
   openssl rsa -in key.pem -out server.key

6. Introduzca la clave creada en el paso 3.

7- Invoca openssl con la función x509:

   openssl x509 in cert.csr -out server.crt -req -signkey server.key -days 60 Ahora, incorporaremos el certificado nuevo y la llave dentro del sistema por defecto en el servidor web.,

8. Copia el certificado dentro del directorio con que cuenta mod_ssl.
   cp -f .server.crt. /etc/httpd/conf /ssl . crt 9-

   Copie la llave dentro del directorio apropiado:

10. Apache necesita ser reiniciado para que los cambios tengan efecto. Debe hacerlo invocando el siguiente comando:
    /etc/init.d/httpd restart

11. Si no esta corriendo el X, inicielo con el comando startx.

12. Cargue Mozilla desdel menu.

13. Escriba la siguiente dirección:
    https: //local host

14. Observe que la información del certificado aparece, entonces haga click en Next.

15. Haga click en el boton More info para ver la información que fué introducida cuando estabas creando el certificado. Haga click en Ok para regresar a la pantalla del certificado.

16. Click en Next, Continue y Ok a traves de las páginas que siguen hasta que cargue finalmente. Note la posicion del padlock's clasp en la esquina izquierda más baja de la pantalla.

Ejercicio 8-4: Seguridad de Apache

Las soluciones a este ejercicio son proporcionados en el Apéndice B.

1. Utilice el acceso basado en el host para prevenir que el computador accese a las páginas man localizadas en tu servidor mientras permite a otros continuar a tener acceso. Esto requerira más de un computador en la red.

2. Utlice la autenticacion de usuario para restringir el acceso al directorio de páginas man en su servidor que creo en el paso anterior. El acceso debería ser solamente concedido al usuario test1.

3. Cree un archivo de autenticacion de usuario en su directorio configuración apache.

4. Detenga y reinicie el servidor.

5. Utilizando su navegador, asegúrese que la seguridad esta trabajando. Pruebe ambos, su configuración de autenticacion de usuario y su configuración acceso basado en host.

6. Visite www.apache-ssl.org y revise sus opciones para utilizar SSL sobre Apache.

7- Lea las páginas del manual de Apache y mire si usted puede implementar el uso de .htaccess.

Ejercicio 8-5: Administración de Apache

Este ejercicio proporcionará la oportunidad de llevar a cabo una sesion corta sobre administrar un servidor web Apache. Las soluciones a este ejercicio son proporcionadas en el Apéndice B.

1. Renombre su index.html y explore su servidor de nuevo. Usted consigue un listado de directorios ?

2. Accione la directiva Fancylndexing en su configuración de Apache. Reinicie Apache para que los cambios tengan efecto. Piensa usted que el indice generado por Apache luce mejor o peor con la directiva DancyIndexing habilitada ?

3. Utilice la directiva IndexOptions para cambiar los contenidos del indice:
   o IconsAreLinks
   :d SuppressSize
   3 SuppressDescription

4. Cree una pagina de myssi.html que utiliza un Server Side Include (SSI) para incluir un archivo sample.txt:

5. Como esto no esta trabajando, usted necesitara alterar la configuración; asi, renomre su arvhivo myssi.shtml e intente probar la pagina de nuevo.

6. Intente algunos otras directivas SSI, tales como:
   #echo #conf1g

7. Revise sus archivos de configuración y anote sus archivos de configuración y anote los nombres y ubicaciones de sus archivos log: los archivos AccessLog y KrrorLog.

8. Revise el contenido actual de estos archivos.

9. Edite el archivo httpd.conf y agregue registros adicionales descomentando las siguientes líneas:
   CustomLog /var/1 og/tittpd/apache/referer_log refercr CustomLog /var/1og/httpd/apache/agent_log agent Stop and restart the server.
   Entre al sitio con el navegador un número de veces con peticiones validas e invalidas y haga clic sobre un número de links. Revise los dos nuevos archivos log; que información que proporciona ?

Ejercicio 8-6: Compilar Apache

En este ejercicio, descargaremos los fuentes de Apache y crear el servidor Web. Si usted quiere permitir algunos de los módulos que su vendedor no incluyo. Necesitarás compilar de la misma manera para hacerlo en este ejercicio. La solución a este ejercicio son proporcionadas en el Apéndice B al final de este manual.

Este ejercicio instalara Apache en el árbol de directorio /usr/local/ , asi que no debería interferir con la instalación existente de Apache, el cual debe estar dentro de /usr/tree.

1. Descargue la última versión de los fuentes de Apache desde ftp.apache.org.

2. Desempaquete el código tarball.

3. Cambie al directorio fuente de Apache y configurelo. (Change to the Apache source directory and configure it. (Debe configurar para instalar /usr/local por defecto.

4. Construya elservidor Apache (Esto tomará algunos minutos crearlo).

5. Instale Apache.

6. Detenga el servicio viejo de Apache.

7. Inicie el nuevo servicio de Apache y asegúrese que trabaja. (Puedes necesitar fijar el campo ServerName en el archivo de configuración.

8. Asegúrese de que usted puede tener acceso al servidor del HTTP en localhost.

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas están el el Apéndice A.

1. Nombre los tres archivos de configuración esenciales de Apache.

2. Que archivo necesitas modificar para permitir o negar acceso a un directorio en tu servidor ?

3. Por que Apache no soporta Secure Sockets Layer (SSL)?

4. Como usted inicia y detiene un servidor Apache?

5. Los servidores se pueden fijar hasta los anfitriones numerosos del servicio. Dé a ejemplo de los directorios del contorno general al sistema para arriba un anfitrión virtual.

CAPÍTULO 9

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Qué es MIME?

2. ¿Cómo es que el SMTP (Simple Mail Transfer Protocol) maneja el correo?

3. ¿Cuáles dos partes conforman una dirección?

4. Describa un host de Correo.

5. ¿Cuál es responsabilidad de Agente de Usuario de Correo (MUA)?

Ejercicio 9-1: Correo/Email

Este ejercicio le ayudará a practicar los conceptos aprendidos de correo en una red TCP/IP. efectuarse un correo totalmente manual. Intentaremos enviar un correo entrando comandos SMTP directamente en el servidor SMTP. No se proveen soluciones a este ejercicio.

Para poder hacer esta practica contactaremos un servidor de correo conocido por el puerto 25 y usaremos comandos SMTP.

Ejecute un telnet y conectese al puerto 25 del servidor de correo. Deberá ver un saludo estandar de SMTP. Escriba comandos SMTP como se muestra en el ejemplo que sigue, substituya su dirección y el mensaje

apropiado:

En este ejemplo usted a conducido una sesión con un SMTP. Afortunadamente que todo este es efectuado por un programa MUA. Pero siempre es bueno hacer este tipo de practica.

Correo anónimo vía telnet

Una manera muy sencilla de enviar correos anónimos sin necesidad de utilizar repetidores es conectarse a un servidor de SMTP a través de Internet, simplemente haciendo un TELNET al puerto 25.

Los pasos a seguir son los siguientes:

1) Localizar un servidor de SMTP lo suficientemente antiguo como para no incluir en las cabeceras del correo que envía la dirección de la máquina que se le conectó. En el ejemplo, llamaremos a esta máquina aaa.bbb.ccc

   Sin duda, éste es el paso más difícil de todos. Sin un servidor así no es posible enviar correos anónimamente. Puedes comprobar los siguientes , a ver si alguno sirve todavía.

2) Busca para ti un nombre cualquiera de máquina (dirección IP), pero que exista. Puede servir uno cualquiera de FTP, HTTP, o lo que sea. Será tu dirección IP falsa, a la que llamaremos xx.yy.zz.

3) Establece una conexión TELNET al puerto 25 con la máquina encontrada en el paso 1.

   En GNU/Linux o ÓNIX, escriba:

   telnet aaa.bbb.ccc 25

4) Si el sitio acepta la petición de conexión, te aparecerá un mensaje como

   220 aaa.bbb.ccc ESMTP Sendmail 8.7.6/8.7.3; Tue, 3 Feb 1998 16:45:30+0100

5) Después de la bienvenida de la máquina, salúdale tú escribiendo:

   HELO xx.yy.zz

   a lo que el host responderá con alguna clase de presentación, como por ejemplo:

   250 aaa.bbb.ccc Hello xx.yy.zz [###.###.###.###], pleased to meet you

6) Escribe los siguientes comandos, sin olvidar el retorno de carro al final de cada línea:

```
MAIL FROM: <unadireccion@falsa.es>
RCPT TO: <tudireccion@verdadera.es>
DATA
Subject: El tema del correo
A continuación el texto del mensaje. No olvides dar un
retorno de carro adicional después del subject. Todos
los mensajes deben terminar con un punto en una línea sola.
.QUIT
```

Con esto ya habrías enviado un correo y cerrado la sesión con el host. Espera a que te llega el mail y examina las cabeceras para ver si ha quedado rastro de tu dirección de máquina o algo que te delate. Si fuera así, vuelve al paso 1 y busca un servidor de SMTP apropiado.

A modo de ejemplo, puedes utilizar el applet de demostración en .

Utilizando el applet, recibirás algo como

Received: from localhost (tu verdadera máquina) by aaa.bbb.ccc (8.7.6/8.7.3) with SMTP id RAA08608 for ; Tue, 3 Feb 1998 17:02:43 +0100

que desvela tu dirección, por lo ese servidor no puede ser usado para enviar correos anónimos. Sin embargo hay otros que sí que lo son. Sólo tienes que buscarlos.

PREGUNTAS POST-EXAMEN

Respuestas a estas preguntas se encuentran en el appendice A.

1.- Si el mensaje enviado es retornado, ¿qué pasos deben ser tomados para determinar el problema?

2.- ¿ Cúal es el propósito de los alias?

3.- ¿ Qué puede un servidor de correo hacer para protegerse de spam?

CAPÍTULO 10

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son los beneficios de usar NFS?

2. ¿Cómo es que un cliente de NFS accesa los directorios de otros equipos?

3. ¿Por qué debe usted 'hard mount' el NFS?

4. ¿Por qué es el comando mount único al sistema operativo?

5. ¿Cuándo debe usted montar su NFS?

Ejercicio 10-1: Crear un Recurso Exportado NFS

En este ejercicio exportaremos el directorio home de una computadora y la montaremos en otra. La máquina que compartirá sus archivos la llamaremos el server, mientras que la máquina que monta el recurso llamaremos el cliente. Será necesario acceso a la cuenta de root para llevar acabo este ejercicio. No se proveen soluciones para este ejercicio.

1.- Especifiquemos que recurso queremos compartir con el cliente. Elegiremos el directorio del usuario completo /home/usuario. Debemos editar el archivo /etc/exports en el servidor y le agregaremos la siguiente línea:

/home/nombre_usuario 192.168.2.7 (rw)

Esta sentencia compartirá el directorio /home/nombre_usuario con el computador cuyo <IP> es 192.168.2.7. Una vez el directorio es montado en el cliente, este tendrá acceso de de lectura y escritura.

2.- Paremos y reiniciemos el servidor NFS:

/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start

3.- Edite el archivo /etc/fstab en el cliente para montar en directorio en el Cliente. Agregue la siguiente línea

en el archivo /etc/fstab en la máquina Cliente:
192.168.2.7:/home/nombre_usuario /mnt/dir_nfs nfs defaults 0 0

4.- Crearemos el directo donde vamos a montar en la máquina Cliente el recurso compartido:
mkdir /mnt/dir_nfs

5.- Montar el recurso compartido en el directorio /mnt/dir_nfs en la máquina Cliente:
mount /mnt/dir_nfs

6.- Navegue al directorio /mnt/dir_nfs en la máquina Cliente a ver si monto correctamente:
cd /mnt/dir_nfs

7.- Cree un archivo nuevo dentro del directorio /mnt/dir_nfs en la máquina Cliente para asegurarse que tiene permisos de lectura y escritura:
touch mi_archivo.txt

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Qué se encuentra en el archivo /etc/exports en el servidor ejecutando NFS?

2.- ¿Cómo funciona el cerrado (locking) de archivos en NFS?

3.- ¿Por qué es necesario el protocolo XDR (External Data Representation)?

4.- ¿Por qué es que el NFS usa UDP?

CAPÍTULO 11

PREGUNTA PRE-EXAMEN

Repuestas a estas preguntas se encuentran en el Apéndice A.

1. ¿Cuáles son las secciones básicas de los archivos de configuración del servidor de Samba?

2. ¿Cúales programas puede usted usar para conectarse con archivos compartidos por Windows?

3. ¿Cúales daemons necesitan estar siendo ejecutados para que el cliente de samba funcione?

4. ¿Puede ser montado un directorio compartido en Windows ser montado en un sistema GNU/Linux?

5. ¿Cuál archivo contiene la descripción completa de los parámetros de Samba?

Ejercicios 11-1: Defina Usuarios de Samba

Debe tener una ID de usuario válida en un dominio de Windows NT. La soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Asegúrese que el archivo /etc/smb.conf esta correctamente establecido para su dominio NT y que Samba

se esta ejecutado.

2.- Use el comando smbadduser para mapear el ID de un usuario en un dominio de Windows NT al equipo local de GNU/Linux usando el mismo nombre de usuario (ahí es necesario deberá crear el usuario).

Ejercicios 11-2: Habilitar swat

Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Verifique que el servicio de inetd/xinetd este configurado para el puerto 901.

2.- Edite el archivo /etc/inetd.conf y elimine el comentario listando el servicio de swat.

3.- Reinicie el daemon de inetd para habilitar el servicio de swat.

4.- Entre al sistema X Window e inicie un navegador web. Escriba la url http://localhost:901/ y entre a la administración de de Samba atraves de swat.

Ejercicios 11-3: Crear un Recurso Compartido de Samba

Para poder llevar acabo este ejercicio es necesario que ya tenga un servidor de samba ejecutándose y una estación de trabajo ejecutando Windows (98 o NT). Si ha seguido todos los pasos hasta el ejercicio anterior entonces ya este debe ser el caso. Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Cree un recurso compartido y llamelo Compartidos apuntelo hacia el directorio /home/usuario/mis_documentos. El nombre del usuario debe ser el mismo usuario que el de la estación de trabajo de Windows NT. Otorguese permisos administrativos a este recurso. El recurso de establecerlo como escribible/writeable y navegable/browseable. Recuerde que en el ejercicios 11-1 se creo este usuario en la estación de trabajo GNU/Linux.

2.- Use el utilitario smbclient en el servidor de Samba para verificar que el recurso está listado efectivamente en los servicios disponibles.

3.- Dirijase a la estación de trabajo ejecutando Windows e ingrese (login) con el usuario que se creo y al cual le compartió el recurso. Use la aplicación gráfica Network Neighborhood para navegar el servidor de GNU/Linux. Verifique que puede navegar e escribir remotamente al recurso compartido.

Ejercicios 11-4: Montar un Recurso Compartido de Windows

Para poder llevar acabo este ejercicio es necesario que ya tenga un servidor de samba ejecutándose y una estación de trabajo ejecutando Windows (98 o NT). Si ha seguido todos los pasos hasta el ejercicio anterior entonces ya este debe ser el caso. Las soluciones a este ejercicio se encuentran en el Apéndice B.

1.- Encuentre un recurso compartido en la estación de trabajo de Windows (NT o 98) que usted tenga otorgado permisos de escritura.

2.- Usando el comando smbmount monte el directorio compartido en el directorio llamada (si no esta creado, creelo) /NT/Compartidos (Compartidos es el nombre del recurso en Windows).

3.- Use comandos de navegación de sistemas de archivos para ver el recurso compartido ya en GNU/Linux. ¿Nota usted alguna diferencia entre la manera de listar de su ext3 y el NTFS?

PREGUNTAS POST-EXAMEN

Las respuestas a estas preguntas se encuentran en el Apéndice A.

1.- ¿Qué protocolos usa el suite de Samba?

2.- ¿ Cúal es el rol de nmbd en Samba?

3.- ¿ Puede el smbclient ser usado para conectarse a otra estación de trabajo GNU/Linux ejecutando Samba?

4.- Escriba el comando que listara todos los recursos compartidos de una estación de trabajo de nombre abiertos.